Fixed Orientation Interconnection Problems: Theory, Algorithms and Applications

Zachariasen, Martin

Publication date:
2009

Document version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Zachariasen, M. (2009). *Fixed Orientation Interconnection Problems: Theory, Algorithms and Applications*. Museum Tusculanum.

FACULTY OF SCIENCE
THE UNIVERSITY OF COPENHAGEN

Martin Zachariasen

Fixed Orientation Interconnection Problems: Theory, Algorithms and Applications

DOCTORAL DISSERTATION

Fixed Orientation Interconnection Problems: Theory, Algorithms and Applications

DOCTORAL DISSERTATION

Martin Zachariasen

Department of Computer Science (DIKU)
Faculty of Science
University of Copenhagen

October 2009

Denne afhandling er i forbindelse med de nedenfor anførte offentliggjorte afhandlinger af Det Naturvidenskabelige Fakultet ved Københavns Universitet antaget til offentligt at forsvares for den naturvidenskabelige doktorgrad.

Forsvaret finder sted den 21. maj 2010 kl. 14.00 i Anneksauditorium A, Studiestræde 6.

København den 15. marts 2010.

Dekan Nils O. Andersen

M. Brazil, B. K. Nielsen, P. Winter, and M. Zachariasen. Rotationally Optimal Spanning and Steiner Trees in Uniform Orientation Metrics. *Computational Geometry: Theory and Applications*, 29:251–263, 2004.

M. Brazil, D. A. Thomas, J. F. Weng, and M. Zachariasen. Canonical Forms and Algorithms for Steiner Trees in Uniform Orientation Metrics. *Algorithmica*, 44:281–300, 2006.

M. Brazil, P. Winter, and M. Zachariasen. Flexibility of Steiner Trees in Uniform Orientation Metrics. *Networks*, 46:142–153, 2005.

M. Brazil and M. Zachariasen. Steiner Trees for Fixed Orientation Metrics. *Journal of Global Optimization*, 43:141–169, 2009.

G. Narasimhan and M. Zachariasen. Geometric Minimum Spanning Trees via Well-Separated Pair Decompositions. *ACM Journal of Experimental Algorithmics*, 6, 2001.

B. K. Nielsen, P. Winter, and M. Zachariasen. An Exact Algorithm for the Uniformly-Oriented Steiner Tree Problem. In *Proceedings of the 10th European Symposium on Algorithms, Lecture Notes in Computer Science*, volume 2461, pages 760–772. Springer, 2002.

B. K. Nielsen, P. Winter, and M. Zachariasen. On the Location of Steiner Points in Uniformly-Oriented Steiner Trees. *Information Processing Letters*, 83:237–241, 2002.

M. Paluszewski, P. Winter, and M. Zachariasen. A New Paradigm for General Architecture Routing. In *Proceedings of the 14th ACM Great Lakes symposium on VLSI (GLSVLSI)*, pages 202–207, 2004.

S. Peyer, M. Zachariasen, and D. G. Jørgensen. Delay-related Secondary Objectives for Rectilinear Steiner Minimum Trees. *Discrete Applied Mathematics*, 136:271–298, 2004.

M. Zachariasen. A Catalog of Hanan Grid Problems. *Networks*, 38:76–83, 2001.

M. Zachariasen. Comment on “Computing the Shortest Network under a Fixed Topology”. *IEEE Transactions on Computers*, 55:783–784, 2006.

M. Zachariasen and A. Rohe. Rectilinear Group Steiner Trees and Applications in VLSI Design. *Mathematical Programming*, 94:407–433, 2003.

Abstract

Interconnection problems have natural applications in the design of integrated circuits (or chips). A modern chip consists of billions of transistors that are connected by metal wires on the surface of the chip. These metal wires are routed on a (fairly small) number of layers in such a way that electrically independent nets do not intersect each other. Traditional manufacturing technology limits the orientations of the wires to be either horizontal or vertical — and is known as Manhattan architecture.

Over the last decade there has been a growing interest in general architectures, where more than two perpendicular orientations can be used for routing. This development has made fixed orientation interconnection problems (where an arbitrary set of fixed orientations can be used) interesting from a research point of view. In particular, the problem of computing minimum length networks with fixed orientations — the so-called fixed orientation Steiner tree problem — has received significant attention.

This doctoral dissertation is a collection of twelve research papers and a survey on the fixed orientation Steiner tree problem and some of its generalizations. One of the main contributions is a linear time algorithm for computing a Steiner minimum tree for a given full topology. Also, a linear programming formulation is presented for the problem. For the general problem an exact algorithm that computes optimal solutions to problem instances with thousands of points is described and implemented. A novel paradigm for routing a chip using a general architecture is implemented and tested on a set of benchmark instances; the approach documents the advantages of using more than two fixed orientations in chip design.

The last part of the dissertation is concerned with generalizations that are motivated by chip design. Firstly, a catalog of problems that can be solved on the so-called Hanan grid is presented. Next, generalizations related to signal delay and group interconnections are studied, and finally, properties of the rotational Steiner tree problem are given.

The results of the dissertation represent a significant step forward, both concerning theory and algorithms, for the fixed orientation Steiner tree problem. In addition, the work maintains a close link to applications and generalizations motivated by chip design.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the doctoral dissertation at the Faculty of Science, University of Copenhagen.

The work of the dissertation developed over the last 10 years. After finishing my Ph.D. in 1998, I continued my research on Steiner trees, slowly moving away from the classical Euclidean and rectilinear versions to various generalizations motivated by chip design. In 1999–2000 I visited the Research Institute for Discrete Mathematics, University of Bonn (headed by Prof. Bernhard Korte) for a period of 7 months; the institute in Bonn was then — and is still — specialized in mathematics and computer science research related to chip design. Furthermore, several of the important contributions of this dissertation were developed during short-term visits to the Department of Electrical and Electronic Engineering, University of Melbourne (headed by Prof. Doreen Thomas) in 2002, 2005 and 2007; my colleagues in Melbourne have also visited Copenhagen on several occations.

The dissertation consists of an introduction to the subject and twelve research papers. The introduction is meant as a moderate level survey for readers with a background in computer science, mathematics or engineering. The focus of the introduction is a balance between the historical development of the subject and major mathematical/algorithmic results. The introduction is comprehensive and attempts to cover all literature on the fixed orientation Steiner tree problem. Open problems are given at the end of the introduction.

The twelve research papers, referred to as [23, 27, 29, 30, 152, 154, 155, 159, 163, 232, 234, 235], are listed among the other references in the introduction (page 71) and separately on page 94. The papers were published from 2001 to 2009, but the bulk of the research was made in the period 1999–2007.

I am most indebted to all my coauthors. In particular I thank my oldest colleague and Ph.D. supervisor, Pawel Winter (coauthor on 5 papers), my Steiner tree friend and colleague in Melbourne, Marcus Brazil (coauthor on 4 papers) and Benny K. Nielsen, former Ph.D. student (coauthor on 3 papers). David Grove Jørgensen, Giri Narasimhan, Martin Paluszweski, Sven Peyer, Andre Rohe, Doreen Thomas, and Jia Weng coauthored one paper each. I thank you all for great collaboration.

I found some great friends and colleagues in Bonn and in Melbourne, and I thank Prof. Bernhard Korte and Prof. Doreen Thomas for making these visits possible.

I thank former librarian Birhe Hougaard for obtaining copies of not so recent literature, and Martin Paluszewski and Sven Peyer for allowing me to use a couple figures produced by them. Marcus Brazil, Julia Lawall and Pawel Winter are thanked for spending time on commenting and proofreading earlier drafts of the introduction to the dissertation. I thank my father, Petur, for help with the graphical design of the front page.

I thank the members of the (now former) Algorithmics and Optimization Group at DIKU for moral support and valuable discussions (in particular my old friends Prof. Emeritus Jakob Krarup and Prof. David Pisinger). Former Heads of Department Stig Skelboe and Jørgen Bansler are thanked for giving me enough research time to write this dissertation. Also, I specially thank Dean Nils O. Andersen, Faculty of Science, for encouraging me to continue my work on the dissertation while being Head of Department.

The management and staff at DIKU is thanked for their patience. In particular I thank Jon Sporring, Ken Friis Larsen, Kim Steenstrup Pedersen, Steen Georg Brandt, Henrik Ingerslev, Inge Hviid Jensen and Lisa Schultz for their support during the last couple of years.

Finally, I thank my wife Elin for her undiminished belief that I would one day finish this dissertation.

Copenhagen, October 2009

Martin Zachariasen

Contents

1 Introduction

In this section we set the scene for the dissertation. We introduce the chip design problem, and in particular the routing problem in physical chip design. The routing problem is one of the main motivations for studying the rectilinear Steiner tree problem, which together with the Euclidean Steiner tree problem form the foundation of the research presented in the dissertation. An overview of important theoretical results and algorithmic developments related to the Euclidean and rectilinear Steiner tree problems concludes the introduction — together with an outline of the complete dissertation.

1.1 Chip design

Chip design — or very-large-scale integration (VLSI) design — is the process of creating complex transistor-based integrated circuits. Chip design consists of several interdependent steps, each of which can be formulated as a huge, and in most cases, NP-hard optimization problem. Due to the size of current designs, where several billion interconnected transistors must be placed on a chip surface, the problem is usually solved hierarchically and broken into a number of (basically) independent steps.

Integrated circuits are fabricated on silicon wafers (also called substrate). By marking different areas of the substrate using photolithography, patterns/tracks consisting of polysilicon, insulator or metal can be deposited on the substrate (Figure 1). In this way transistors and wires connecting them can be built on a very small scale on the surface of the substrate. Current technology makes it possible to construct patterns/tracks less than 50 nanometers wide.

The construction of an integrated chip begins by describing the behavior of the chip. *Logic synthesis* is the process of specifying the logic functions of the chip and their interrelations. It is done using a hardware description language (HDL). *Logic optimization* turns the description into a compact and efficient — but logically equivalent — description. The result is a *netlist*, which describes how a set of standard components such as NANDs or NORs are interconnected. The final step of logic synthesis is the mapping of each standard component to a specific implementation (or physical drawing); the choice of implementation depends on requirements related to area consumption, load capacitance and timing. The re-

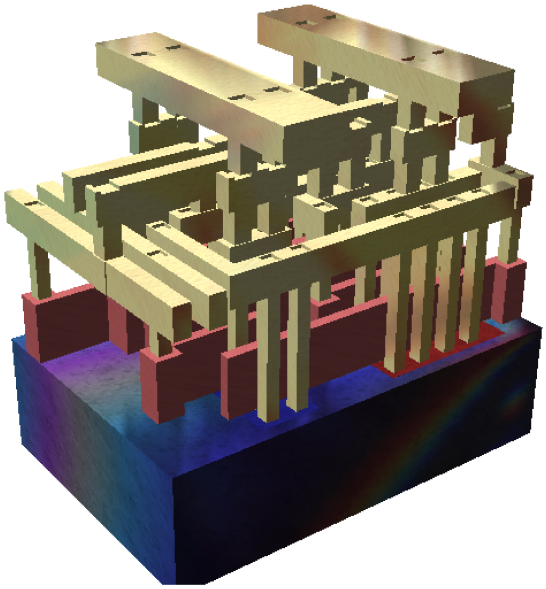


Figure 1: Small integrated circuit with three metal layers (insulator has been removed). The sand-colored structures at the top are metal interconnect. The layers are connected using vias (vertical pillars). The reddish middle structures are polysilicon gates, and the solid at the bottom is the substrate.

sult is a netlist where each *net* interconnects a set of *modules* (or *cells*) — each of which has a given physical realization.

The process of locating the modules and wires on the chip surface such that, e.g., area usage and signal delay is minimized, is called *physical design*. The first step of physical design is usually *floorplanning*, where major parts of the circuit are placed on the chip surface. (For an integrated circuit of a CPU, such major parts could be arithmetic logic unit, branch predictor, cache etc.) In the *placement* step each of the modules is located on the chip surface. The placement problem is a multi-objective problem, where area usage, wire length and signal delay are the primary objectives. These objectives are usually combined into a single quality measure called *netlength*. The problem of meeting timing (or clock rate) constraints is called *timing optimization*, and is often performed by adjusting the netlength of critical nets and reoptimizing the placement under the new netlength objective.

The final step of physical design is *routing*, where the wires interconnecting the

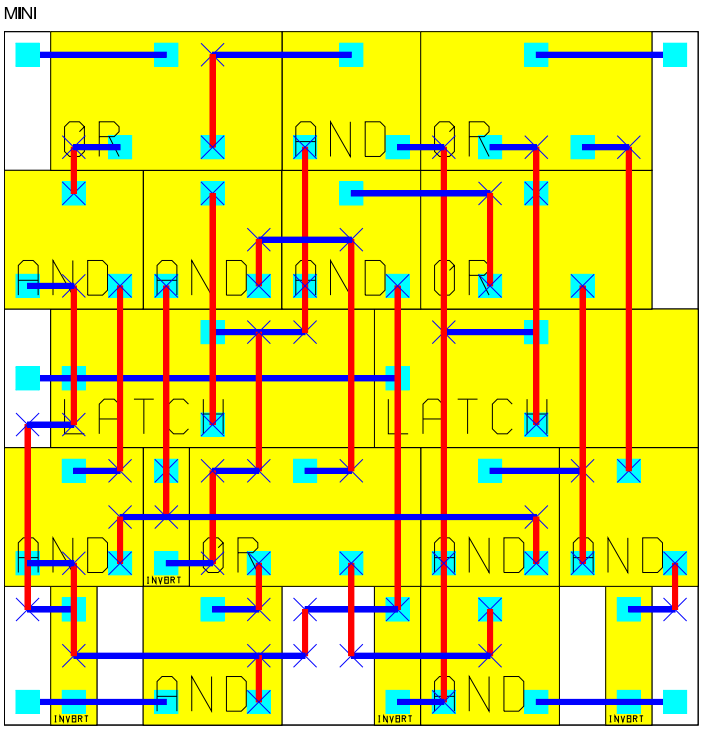


Figure 2: A placement and a routing of a small circuit. The circuit has 19 modules, 22 nets and 58 pins. The horizontal blue wires and vertical red wires run on different metal layers. Vias are indicated by an "X". (Figure reproduced by courtesy of Research Institute for Discrete Mathematics, University of Bonn.)

modules are located on the chip surface and assigned to different (metal) layers. Wires on a given layer have a preferred direction which is either horizontal or vertical (Manhattan routing). Each net of the netlist should interconnect a given set of *pins/terminals* on the chip surface, and wires from different nets should not intersect each other. The main objective of routing is firstly to obtain a feasible routing and secondly, to minimize total wire length and signal delay. Figure 2 illustrates a small chip where placement and routing has been performed.

Routing is performed with the help of a (flat) three-dimensional grid graph, where the distance between neighbouring grid lines is the minimum width of a wire plus the minimum distance between wires. In older technologies modules and wires were aligned perfectly on the grid graph, but in current designs, modules and wires can be placed *off-grid*. Modern routing algorithms therefore only implicitly use the grid graph.

Routing millions of nets in a grid graph with billions of nodes is a challenging task. The problem is therefore divided into at least two steps: *global routing* and *detailled routing*. The global routing problem is a coarse version of the routing problem, where the chip surface is divided into axis-aliged rectangular regions. The height and width of a region is typically 50–100 grid lines in the grid graph. In the *global routing grid graph* the vertices are the regions, and two vertices are connected by an edge if the two regions are neighbours. The edges in the global routing grid graph have associated lengths and capacities, where the capacities estimate the maximum number of wires than can be routed between two neighbouring regions. The global routing problem is, in its simplest form, a so-called Steiner tree packing problem, where trees should be "packed" in the global routing grid graph such that the capacities of the edges are respected.

The output of the global routing problem is a "global routing corridor" for each nets, that is, a coarse description of the wiring of each net. In detailed routing the exact wiring of each net is determined — and in such a way that the output of global routing is respected. Using the output from global routing both minimizes the risk of congestion in detailed routing and speeds up detailed routing, since only a relatively small part of the full grid graph needs to be considered. Due to the size of the detailed routing problem, the problem is normally solved one net at at time — and in most cases one point-to-point connection at at time. While global routing is a multi-objective problem where congestion, timing and wire length is considered, detailed routing is primarily concerned with feasibility, and the only real optimization involved is (implicit) shortest-path computation in the grid graph.

The literature on physical design of integrated circuits is vast. Some fairly recent books and theses include — in chronological order — Lengauer [126], Kahng and Robins [113], Pecht and Wong [160], Sarrafzadeh and Wong [181], Gerez [83], Sait and Youssef [175], Sherwani [185], Vygen [204] and Saxena et al. [182]. An early tutorial on the routing problem is Hightower [94], and more recent surveys can be found in Möhring et al. [147], Cong et al. [63] and Peyer [162].

The list of combinatorial problems in chip design recently compiled by Korte and Vygen [117] illustrates the challenges in the field. The authors consider the chip design problem to be one of the most important application areas in (discrete) mathematics. In particular, efficient algorithms are needed to handle problems with millions of modules and nets.

1.2 Steiner tree problem

The routing problem in chip design motivates the study of interconnection problems in the plane. In this section we introduce the *Steiner tree problem* — the problem of interconnecting a given set of points in the plane by a tree of minimum length. Algorithms for this problem play a fundamental role in chip design routing. We briefly present the history and research development on the major variants of the Steiner tree problem, namely the *Euclidean* Steiner tree problem, the *rectilinear* Steiner tree problem and the Steiner tree problem in *graphs*.

History of the Euclidean Steiner tree problem

The roots of the Euclidean Steiner tree problem go back to Fermat early in the 17th century [118, 229]. The problem was presented as a challenge in a celebrated essay on maxima and minima: "Let he who does not approve of my method attempt the solution to the following problem: Given three points in the plane, find a fourth point such that the sum of its distances to the three given points is a minimum!"

Torricelli proposed a geometric solution to the problem before 1640 by constructing equilateral triangles and corresponding circumscribing circles on the sides of and outside the given triangle. The circumscribing circles intersect at the fourth point, which today is called the Fermat-Torricelli point — or the Steiner-Weber point or just the Steiner point — of the given three points (Figure 3).

The Fermat problem can be generalized in several ways. One is to allow more than three given points, but still consider the problem of finding a single point that minimizes the sum of distances to the given points. Furthermore, each of the distances can be weighted with some positive number. This problem is called the *general* Fermat problem; a nice overview of properties related to this problem, including some elegant duality results, are presented by Kuhn [118].

Another generalization is the *Steiner tree problem* — the main topic of this dissertation. Here a set N of n points are given in the plane, and the problem is to compute a shortest network that interconnects these points. (Note that such a network will always be a tree, and that the Fermat problem is the special case where $n = 3$.) The Steiner tree problem appears to have been suggested for the first time in 1934 by Jarník and Kössler [107], and a famous mathematics book by Courant and Robbins [69] gave the problem its name. Although Jakob Steiner was a well-

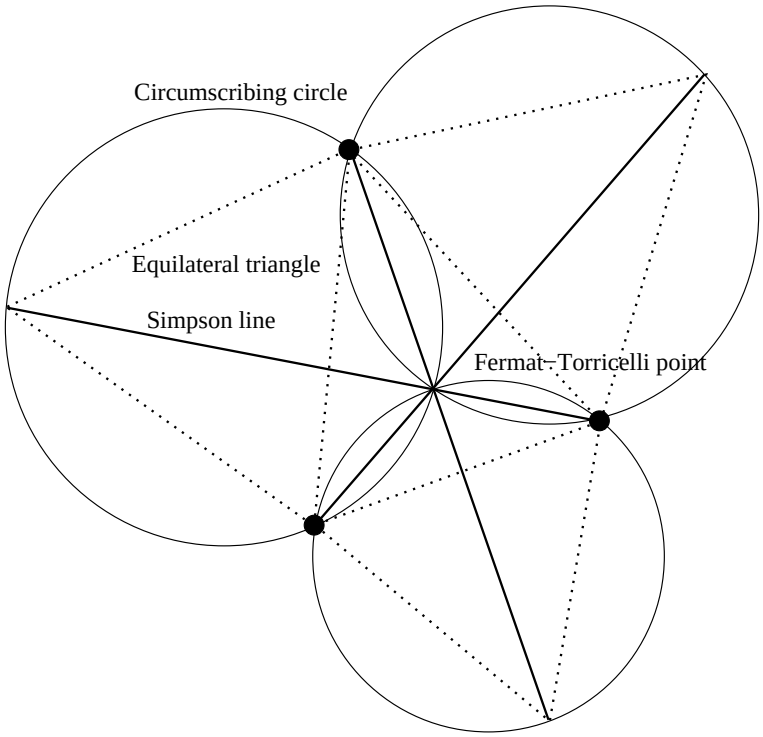


Figure 3: Construction of the Fermat-Torricelli point of three given points. The circles circumscribing the equilateral triangles on the sides of and outside the given triangle intersect at the Fermat-Torricelli point. Cavalieri argued that the segments connecting the given points with the Fermat-Torricelli point meet at 120° angles. The fact that the Simpson lines, which connect the third corner of the equilateral triangle to the opposite given point, also meet at the Fermat-Torricelli point was shown in 1750 by Simpson. In 1834 Heinen proved that the length of the Simpson lines are equal to the sum of distances from the given points to the Fermat-Torricelli point.

known geometer in the 19th century, he made no significant contributions to the problem.

The first breakthrough on the algorithmic side was made in 1961 by Melzak [145], who gave a finite construction for the problem of computing a Euclidean Steiner tree for a given tree structure (or topology) — resulting in a finite-time algorithm for solving the problem. Several theoretical contributions and generalizations to higher dimensional spaces were given in 1968 by Gilbert and Pollak [85]; they also coined the name *Steiner points* for the vertices in the shortest tree that are not among the given points. The problem reached the general public in 1989 through a popular paper in Scientific American written by Bern and Graham [10].

Rectilinear Steiner tree problem and the Hanan grid

Returning to the routing problem in chip design, recall that the fundamental problem is to connect a set N of *terminals* using a minimum amount of wire. However, due to manufacturing constraints in traditional chip technologies, wires can only run in horizontal and vertical orientations (Manhattan routing). Ignoring the assignment of wires to (metal) layers, this correspond to the *rectilinear Steiner tree problem* — or the problem of computing a minimum length network under the L_1 metric.

Already in 1966 Hanan [88] presented the first thorough study of the rectilinear Steiner tree problem. One of Hanan's key contributions was to show that there exists an SMT in the *Hanan grid* defined by the terminals. The Hanan grid for the terminal set N is obtained by drawing horizontal and vertical lines through each point in N. Correspondingly, the *Hanan grid graph* $\mathbf{GG}(N)$ is defined as follows: The set of intersections in the Hanan grid are the vertices, and a pair of vertices is connected if and only if the corresponding intersection points are adjacent in the Hanan grid. The weight of an edge in $\mathbf{GG}(N)$ is the (Euclidean) distance between the corresponding Hanan grid intersections. Computing a tree of minimum total edge-weight in $\mathbf{GG}(N)$ that interconnects the vertices in N is the same as solving the rectilinear Steiner tree problem (Figure 4).

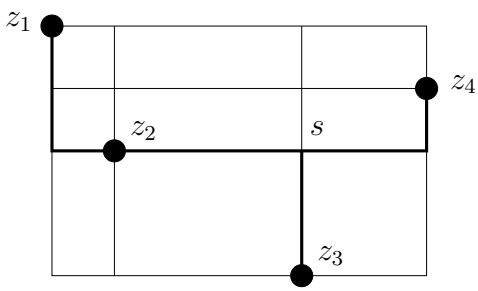


Figure 4: Hanan grid example for $n = 4$ terminals (only line segments within the bounding rectangle are drawn). A Steiner minimum tree (SMT) is drawn with bold lines. Note that the single Steiner point s shares coordinates with the terminals z_2 and z_3.

The Hanan grid graph Steiner tree problem is a special instance of the more general *Steiner tree problem in graphs* [87]: Given an undirected graph $G = (V, E)$ with positive edge-weights and a non-empty set $N \subseteq V$ of terminals, find a minimum edge-weight tree in G that interconnects N. Note that the edge-weights need not be related to any familiar distance metric.

The rectilinear Steiner tree problem can therefore be solved as a graph problem with at most n^2 vertices and $2n(n-1)$ edges, where $n = |N|$. Early algorithms for the general graph problem include the spanning tree enumeration algorithm of Hakimi [87] and the dynamic programming algorithm of Dreyfus and Wagner [70]. More recent algorithms use integer programming formulations which are solved by branch-and-cut [164, 165]. Despite of the existence of efficient reduction methods [202, 218], solving the rectilinear Steiner tree problem via the Hanan grid graph is not competitive with the so-called Geosteiner approach (see Section 1.3).

On the other hand, a number of generalizations of the rectilinear Steiner tree problem can be solved in the underlying Hanan grid. Ganley and Cohoon [78] showed that the rectilinear Steiner tree problem with rectilinear obstacles can be solved in the Hanan grid given by the terminals and the corners of the obstacles. Zachariasen [232] presented a catalog of problems that have an optimal solution in the Hanan grid, including so-called weighted-obstacle, group and prize-collecting variants. Snyder [189] generalized Hanan's results to higher dimensional spaces, and Du and Hwang [73] generalized the result further to any d-dimensional normed space with a unit sphere that is a symmetric polytope with $2d$ extreme points (see Section 2 for definitions).

1.3 Fundamental properties and algorithmic developments

In this section we define some notation and sketch the algorithmic developments related to the Steiner tree problem. Further details can be found in the following books, tutorials and surveys (in chronological order): Hwang and Richards [100], Hwang, Richards and Winter [101], Ivanov and Tuzhilin [106], Cieslik [54], Harris [89], Prömel and Steger [168] and Cieslik [56].

Full topologies and fulsome Steiner trees

The graph structure of a tree T (i.e., the pattern of adjacencies of the vertices for a given labelling of the terminals) is referred to as its *topology* $\mathcal{T}$. A tree T or its topology $\mathcal{T}$ is said to be *full* if all its terminals have degree 1; if every Steiner point furthermore has degree 3, then $\mathcal{T}$ is called a *full Steiner topology*.

We define a *Steiner tree* to be a tree that cannot be shortened by any (small) pertur-

bation of its Steiner points. A shortest tree with a given topology is called a *locally minimal* tree. A locally minimal tree — which by definition is a Steiner tree — can be found efficiently since the minimization problem is convex in the locations of the Steiner points (a local minimum is also a global minimum). We return in Section 3.1 to the problem of computing a locally minimal tree, or abusing the notation slightly, an *SMT for a given topology*.

A locally minimal tree having a full Steiner topology is called a *full Steiner tree (FST)*. In 1967 Cockayne [59] made the observation that a Euclidean SMT is a union of FSTs (Figure 5). Cockayne also gave an algorithm for enumerating full Steiner topologies by showing a connection to balanced parenthesis structures.

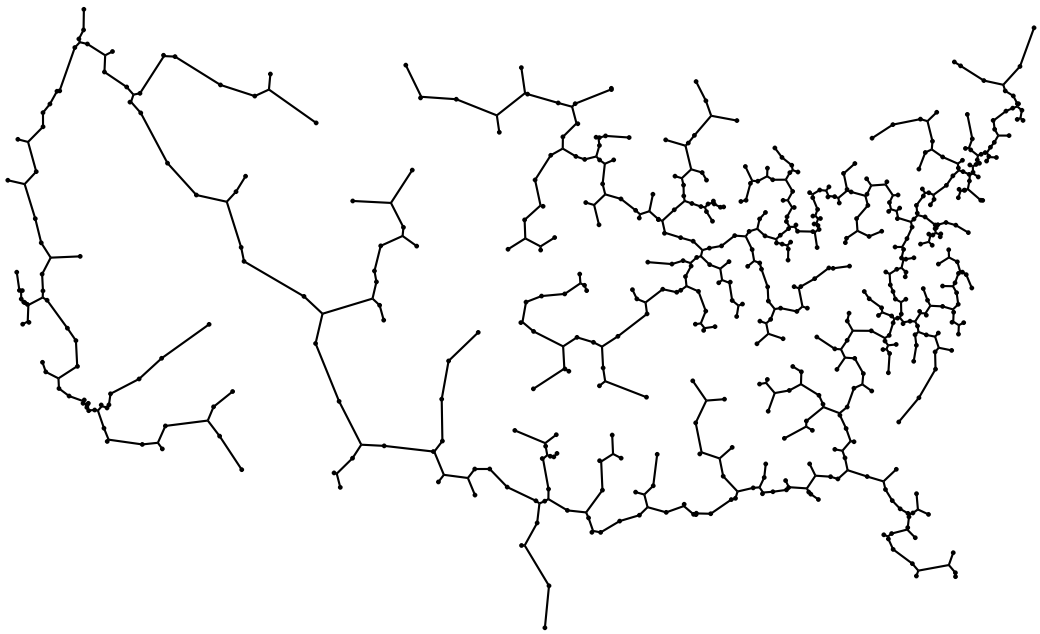


Figure 5: A Euclidean Steiner minimum tree (SMT) for a set of points representing 532 cities in the United States. Note that most full Steiner trees (FSTs) span 2, 3 and 4 terminals.

A tree T or its topology $\mathcal{T}$ is said to be *fulsome* if T contains the maximum possible number of full Steiner trees for any SMT on the terminal set of T. If we perturb the Steiner points in a fulsome SMT (without changing the length of the tree), then we cannot make a Steiner point coincide in position with one of the terminals. Clearly, for any set of terminals there always exists an SMT in which every FST is fulsome.

The notion of fulsome FSTs plays an important role when proving structural properties of *rectilinear* SMTs. Basically, fulsomeness limits the set of SMTs that needs to be considered, and this makes it possible to provide more powerful characterizations. Hwang [97] proved that a rectilinear FST can be assumed to have a certain canonical form where all the terminals are connected in a chain-like fashion to a backbone — the so-called Hwang-topology [231, 233].

Minimum spanning trees and the Steiner ratio

A minimum spanning tree (MST) for a set N of n points in the plane is a shortest network which spans N and does *not* introduce Steiner points. Minimum spanning trees in edge-weighted graphs can be computed in polynomial time (essentially in linear time in the number of edges [66]); by constructing the complete graph on the set of points N, a MST in the plane can be computed in $O(n^2)$ time using, e.g., Prim's algorithm for the corresponding graph problem.

By exploiting the geometry of the problem, Euclidean and rectilinear MSTs in the plane can be constructed in $O(n \log n)$ time [98, 167]. One classical approach is to use the dual of the Voronoi diagram for N, the Delaunay triangulation, to identify a subgraph with $O(n)$ edges of the complete graph on N that contains a MST for N. Another possibility is to use the well-separated pair decomposition data structure [34]; Narasimhan and Zachariasen [152] showed that this data structure allows the efficient computation of Euclidean and rectilinear MSTs in the plane and in higher dimensions. This approach further has the advantage that only minor adaptions are needed to handle new metrics.

Clearly, SMTs are in general shorter than MSTs for the same terminal set N, since SMTs are allowed to contain Steiner points. The infimum over all terminal sets N of the ratio between the SMT length of N and MST length of N is denoted the *Steiner ratio*. Gilbert and Pollak [85] conjectured in 1968 that this value is $\sqrt{3}/2$ for the Euclidean problem. A proof of this conjecture was given by Du and Hwang in 1992 [72]. The variational approach of Rubinstein and Thomas [174] played an important role in proving the Steiner ratio conjecture.

The Steiner ratio for the rectilinear problem is $2/3$ [97]; hence, a rectilinear SMT is at most 33.3% shorter than a corresponding rectilinear MST. A Euclidean SMT is at most 13.4% shorter than a corresponding Euclidean MST. Steiner trees under the rectilinear metric thus have a relatively greater length improvement potential.

Hardness, approximation and exact algorthms

The Euclidean and rectilinear Steiner tree problems were among the early problems that were shown to be NP-hard — and thus essentially eliminating the hope that polynomial-time algorithms exist for the problems. Garey, Graham and Johnson [80] proved in 1977 that there exists no fully polynomial-time approximation scheme (FPTAS) for the Euclidean problem unless P = NP. A FPTAS is an algorithm that for every fixed $\epsilon > 0$ computes a $(1 + \epsilon)$-approximation in polynomial time in the length of the input and $1/\epsilon$. Even though the Euclidean problem is NP-hard, the decision version is not known to be in NP. On the other hand, as a result of the Hanan grid property, the decision version of the rectilinear problem is NP-complete [81].

A polynomial-time approximation scheme (PTAS) is an algorithm that for every fixed $\epsilon > 0$ computes a $(1 + \epsilon)$-approximation in polynomial time in the length of the input (but not necessarily in $1/\epsilon$). It was for a long time a major open problem whether there existed a PTAS for the Steiner tree problem and other geometric problems in the plane. In contrast, the Steiner tree problem in *graphs* was known to be APX-complete, and hence does not admit a PTAS unless P = NP; the best approximation ratio is currently 1.55 [173].

In 1996 Arora [4] gave the first PTAS for the Euclidean Steiner tree problem in the plane — basically as a corollary of a similar result for the Euclidean traveling salesman problem in the plane. The running time of Arora's original algorithm has later been improved by Arora [5] and Rao and Smith [170]. The Rao-Smith algorithm is optimal in the sense that the asymptotic running time bound matches the $\Omega(n \log n)$ lower bound for the algebraic computation tree model.

Motivated primarily by applications in chip design, a range of heuristics have been proposed both for the Euclidean and rectilinear Steiner tree problems in the plane. An overview is given in the book by Hwang, Richards and Winter [101]; also, the papers [48, 49, 111, 236] document some of the more recent and successful contributions.

The work on super-polynomial time exact algorithms has also been substantial. Already in 1970, Cockayne [60] presented one of the first implementations of an exact algorithm for the Euclidean problem. Since then, the so-called Geosteiner approach suggested by Winter [216] in 1985 has by far proved to be the most succesful [61, 62, 89, 207, 208, 209, 220, 231]. The idea is to enumerate full Steiner

trees (FSTs) followed by concatenation into a complete SMT. Today, Euclidean and rectilinear Steiner tree problem instances with several thousand terminals can be solved to optimality.

1.4 Overview of introduction to dissertation

The purpose of the introduction to the dissertation is to give a survey on the literature on the fixed orientation Steiner tree problem. The survey is intended to be self-contained and comprehensive, but not particularly technical. The twelve research papers that form the body of the dissertation are included in a natural way, albeit with some emphasis on the results from the twelve papers.

We begin by surveying known results on structural properties in Section 2. Then we move on to present algorithmic developments for constructing a Steiner tree for a given topology and for the general case in Section 3. Finally, in Section 4, we present applications in chip design and various generalizations motivated by chip design — and in particular contributions related to the *general* fixed orientation problem. Each section ends with a conclusion that summarizes the existing work and future challenges.

2 Theory

In this section we define the uniform and fixed orientation Steiner tree problems and present some of the fundamental properties that are known for these problems. These properties are mainly related to structural properties of optimal solutions, e.g., properties related to angles and directions of edges in an optimal solution. Also, the existence of optimal solutions in the so-called generalized Hanan grid is discussed.

2.1 Uniform and fixed orientation Steiner tree problem

The generalization of the Euclidean Steiner tree problem to other metrics in the plane was discussed in some early works of Melzak [145] and Cockayne [59]. One generalization is to let the distance between two points depend on the orientation of the line segment between the points. Here the orientation of a line (segment) is the angle α it makes with the x-axis (where $0 \leq \alpha < \pi$).

In a *normed* (or Minkowski) plane the ordinary (Euclidean) unit circle is generalized to be the boundary $\mathcal{C} = \partial\mathcal{D}$ of any compact, convex and centrally symmetric domain $\mathcal{D}$. The distance between two points p and q is obtained by placing the center of $\mathcal{C}$ at p, drawing a ray r with base in p and containing q, and identifying the intersection q' of r with $\mathcal{C}$. The distance between p and q is now $|pq|_{\mathcal{C}} = |pq|/|pq'|$, where $|\cdot|$ as usual denotes Euclidean distance. (Note that the points on the unit circle $\mathcal{C}$ in fact have distance 1 from the center of $\mathcal{C}$.)

A normed plane is *strictly convex* if $\mathcal{C}$ is strictly convex, i.e., if the line passing through any pair of points x and y on $\mathcal{C}$ meets $\mathcal{C}$ only at x and y. Equivalently, the triangle inequality is strict for non colinear points in a strictly convex normed plane. A *smooth* normed plane is one for which each boundary point on $\mathcal{C}$ has a unique tangent. The well-known L_p metrics are strictly convex and smooth for $1 < p < \infty$, but the rectilinear metric (L_1) is neither strictly convex nor smooth (Figure 6a).

In this section we consider the Steiner tree problem in normed planes where $\mathcal{C}$ is a centrally symmetric *polygon*. The best known example of such a plane is the rectilinear plane. An important generalization of the rectilinear plane is the λ-geometry plane, in which $\mathcal{C}$ is a regular 2λ-gon for some integer $\lambda \geq 2$ (Figure 6b): the corresponding metric is also denoted the *uniform orientation* metric [179].

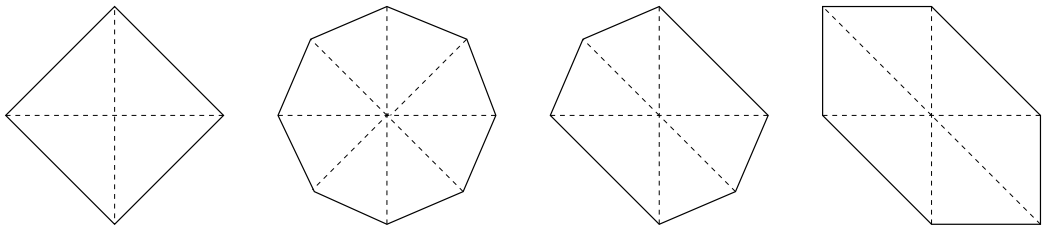


Figure 6: Examples of polygonal unit circles. a) Rectilinear (2 perpendicular orientations). b) Uniform orientation metric ($\lambda = 4$). c) Fixed orientation metric ($\sigma = 3$). d) Weighted fixed orientation metric.

When $\mathcal{C}$ is an arbitrary polygon whose 2σ vertices are inscribed in a Euclidean unit circle, we have the *fixed orientation* metric, or (unweighted) σ-geometry plane (Figure 6c). Finally, in the *weighted fixed orientation* metric (or weighted σ-geometry plane), there are no restrictions on the locations of vertices of $\mathcal{C}$ — apart from $\mathcal{C}$ being convex and centrally symmetric (Figure 6d).

The directions given by the vectors pointing from the center to the 2σ vertices of $\mathcal{C}$ are called *legal directions*. The orientations given by (pairs of opposite) vertices of $\mathcal{C}$ are denoted *legal orientations*. In the first systematic treatment in the algorithmics literature on geometric problems related to (unweighted) fixed orientation metrics, Widmayer et al. [214, 215] defined the fixed orientation metric using legal orientations: The distance between p and q is the shortest geometric path consisting of line segments in legal orientations only. If the orientation of the line through p and q coincides with a legal orientation, then the shortest path is simply a straight line between p and q. In this case we say that the connection between p and q is a *straight edge*. (In the definition of $|pq|_{\mathcal{C}}$ given above this corresponds to the case where the point q' coincides with a vertex of $\mathcal{C}$.) On the other hand, if the orientation of the line through p and q does not coincide with a legal orientation, then any zigzag-path consisting of line segments having the immediate preceding and succeeding legal orientations will constitute a shortest path [215] (Figure 7a and 7b). In fact, such a shortest path connection, or *bent edge*, can be assumed to consist of at most two line segments (having legal orientations) joined by a *corner point* (Figure 7c).

Given a polygonal unit circle $\mathcal{C}$ and a set of terminals N, the *fixed orientation Steiner tree problem* is to construct a shortest possible interconnection of the

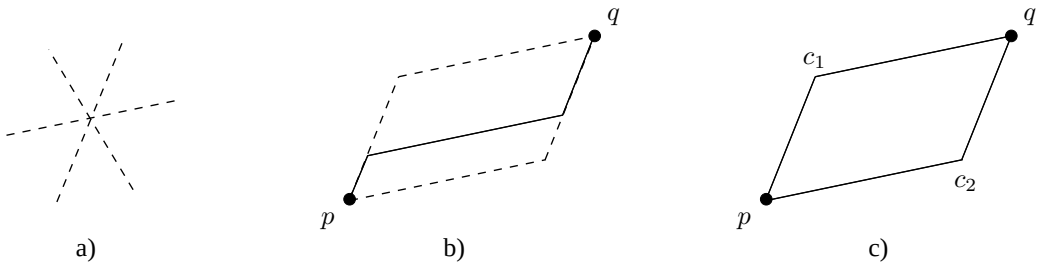


Figure 7: Distances in fixed orientation metrics. a) Given legal orientations. b) Zigzag-path between two points p and q consisting of line segments in legal orientations (here a so-called Z-shaped zigzag-path). c) Bent edges between p and q consisting of exactly two line segments, one containing corner point c_1 and the other containing corner point c_2.

points in N under the metric given by $\mathcal{C}$ (or in the σ-geometry plane). In the uniform orientation metric (or λ-geometry plane), the problem is naturally called the *uniform orientation Steiner tree problem*. As usual, an optimal solution is denoted a *Steiner minimum tree (SMT)*. The Steiner tree problem has been shown to be NP-hard both in the rectilinear plane ($\lambda = 2$) [81] and in the octilinear plane ($\lambda = 4$) [186]. Brazil [22] surveys the developments for the uniform orientation problem up to 2001.

2.2 Fundamental structural properties

One particular difficulty of the Steiner tree problem is that an SMT may contain junctions, or Steiner points, that are not among the given terminals. Consider a Steiner point s with neighbours $x_1, \ldots, x_m$ in an SMT (where $m \geq 3$). The Steiner point is an optimal solution to the general Fermat problem for the points $x_1, \ldots, x_m$ — also called a Fermat-Torricelli point (see Section 1.2). The general Fermat problem has a long history and is of particular interest in location science [118]. A nice survey on results pertaining to this problem in general normed planes and spaces is given by Martin et al. [144].

In this section we first present bounds on the degrees and angles for Steiner points, both for general weighted fixed orientation metrics and for special cases of the metric. Related degree and angle bounds can be obtained for terminals, but from an algorithmic point of view, properties related to Steiner points are substantially more interesting.

The second topic of this section is the so-called centroid property, which both is a theoretically beautiful, but also algorithmically useful property of solutions to the Fermat problem. Then we describe zero-shifts, which are length-preserving perturbations of Steiner points in SMTs. Zero-shifts are particularly interesting for fixed orientation metrics, since these are not strictly convex, and therefore generally result in an infinite set of SMTs for a given set of terminals. Finally, at the end of the section we briefly survey results on the Steiner ratio for fixed orientation metrics.

Degree and angle conditions

A *Steiner configuration* is a star with center s and leaves $x_1, \ldots, x_m$ that is part of some SMT with s as Steiner point (where $s, x_1, \ldots, x_m$ are distinct) . Note that in a Steiner configuration, the center s is a Fermat-Torricelli point for $x_1, \ldots, x_m$. For unit circle $\mathcal{C}$, let $s(\mathcal{C})$ denote the maximum degree of a Steiner point; hence under the metric given by $\mathcal{C}$, the size m of any Steiner configuration is $3 \leq m \leq s(\mathcal{C})$, since Steiner points by definition have degree at least 3.

Theorem 2.1 *For any unit circle $\mathcal{C}$ we have $s(\mathcal{C}) \leq 4$, and for a* smooth *unit circle $\mathcal{C}$ we have $s(\mathcal{C}) = 3$.*

For the (smooth) Euclidean metric the bound of 3 follows immediately from the fact that edges meet at 120° angles, and for the L_1 metric the bound of 4 was shown already by Hanan [88] (a cross forms a degree 4 Steiner configuration). Cockayne [59] proved that $s(\mathcal{C}) = 3$ for smooth and strictly convex unit circles, and Alfaro et al. [2] showed that if the unit circle is strictly convex (but not necessarily smooth), then $s(\mathcal{C}) \leq 4$. (Liu and Du [139] independently proved that $s(\mathcal{C}) = 3$ for all smooth and strictly convex L_p metrics, that is, when $1 < p < \infty$.) Cieslik [52] gave a simple proof of the fact that for arbitrary unit circles (and every terminal set) there always *exists* an SMT for which $s(\mathcal{C}) \leq 4$. Lawlor and Morgan [119] showed that $s(\mathcal{C}) = 3$ for *all* smooth unit circles. Finally, Swanepoel [193] proved Theorem 2.1 in full generality, and in addition, he gave a precise characterization of unit circles that permit a degree 4 Steiner configuration (so-called X-planes).

Since polygonal unit circles are neither smooth nor strictly convex, all we can infer from Theorem 2.1 is that Steiner points have degree 3 or 4. In order to

characterize this more precisely, we now consider the meeting angles at Steiner points for the uniform orientation metric (or in λ-geometry). Let $\omega = \pi/\lambda$ be the angle between neighboring orientations (e.g. for $\lambda = 4$ we have $\omega = \pi/4 = 45°$).

Theorem 2.2 *In λ-geometry, the minimum meeting angle at a Steiner point is $\lceil 2\lambda/3 - 1 \rceil \omega$, while the maximum meeting angle is $\lfloor 2\lambda/3 + 1 \rfloor \omega$.*

Note that meeting angles in λ-geometry are somewhat poorly defined, since one may say that they depend on the actual embedding of the edges. However, the above theorem holds both when edges are considered to be straight line segments between their endpoints, and when they are embedded using legal orientations only.

The first proof of Theorem 2.2 was given by Sarrafzadeh and Wong [180], albeit not covering the case where λ is multiple of 3 correctly. Alternative (and correct proofs) using various proof techniques were given by Koh [115] (only for $\lambda = 4$), Li et al. [129] (only lower bounds), Brazil et al. [24], Swanepoel [193], Hayase [90], and Il'yutko [105].

Since $2\lambda/3 \cdot \omega = 2\pi/3 = 120°$, Theorem 2.2 states that the minimum angle is the largest multiple of ω that is strictly less than $120°$; and similarly, that the maximum angle is the smallest multiple of ω that is strictly greater than $120°$. Thus, as $\lambda \to \infty$, the minimum and maximum angles both converge toward $120°$ as could be expected.

Obviously, if the minimum angle $\lceil 2\lambda/3 - 1 \rceil \omega$ is strictly greater than $90°$ then the maximum Steiner point degree is 3. Direct computation shows that degree 4 Steiner points are only possible for $\lambda = 2, 3, 4$ and 6. Thus we have the following:

Corollary 2.3 *In λ-geometry, Steiner points have degree 3, except when $\lambda = 2, 3, 4$ and 6 (where Steiner points with degree 4 exist).*

If we think of edges as embedded using legal orientations, then for $\lambda = 3k$, a meeting angle can either be $2\pi/3 - \omega$, $2\pi/3$ or $2\pi/3 + \omega$, while for $\lambda \neq 3k$, only two meeting angles are possible. In some sense there is a greater flexibility when λ is a multiple of 3 — something that will be discussed in further detail in Section 3.2.

The lower bound on meeting angles in Theorem 2.2 is still valid for terminals, but the upper bound is not relevant as terminals can have degree less than 3 in an

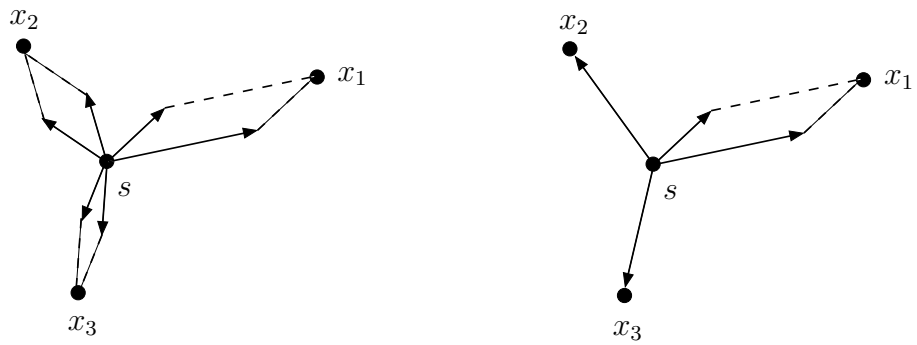


Figure 8: Direction sets with 6 directions (left) and 4 directions (right).

SMT. For general unit circles the *degree* upper bound for terminals is the same as for Steiner points, *except* in λ-geometry for $\lambda = 3$; in this case a terminal can have 6 neighbours (star with six edges separated by $60°$) [52, 193].

Degree properties of Steiner points and terminals for SMTs in higher-dimensional spaces have been studied by, e.g., Cieslik [53] and Swanepoel [192, 194]. One of the interesting results is that for arbitrary smooth unit balls in d-dimensional spaces, the maximum degree of Steiner points and terminals is $d+1$. On the other hand, for the non-smooth L_∞ metric, the maximum degree of both Steiner points and terminals is 2^d, so polyhedral unit balls significantly increase the number of possible Steiner tree topologies.

Direction sets

Consider a Steiner point s with three neighbours x_1, x_2 and x_3 in some SMT (in σ-geometry). If we think of the edges sx_1, sx_2 and sx_3 as pointing outwards from s, then the line segments of each edge use one of the legal directions. Consider the set of legal directions used by all three edges of the corresponding Steiner configuration $\mathcal{S}$ (Figure 8). A Steiner configuration $\mathcal{S}$ is *maximal* if the set of legal directions used by $\mathcal{S}$ does not form a strict subset of the directions used by some other Steiner configuration. We define a *direction set* to be the set of legal directions used by some maximal Steiner configuration [30].

Each edge in a Steiner configuration contributes to a direction set with one direction (straight edge) *or* two adjacent directions (bent edge). Thus a direction set contains from 3 to 6 directions. It turns out that a direction set with 3 direc-

tions (corresponding to a Steiner configuration with straight edges only) cannot exist: such a direction set can always be supplemented with at least one direction, hence contradicting maximality [30]. Therefore, direction sets contain from 4 to 6 directions.

If an edge in a Steiner configuration contributes two (adjacent) directions to a direction set, the first one (in counter-clockwise order) is denoted a *primary* direction while the other is denoted a *secondary* direction. Single-direction edges can be labeled either primary or secondary.

Whenever an edge contributes two directions these are adjacent, so we immediately obtain an upper bound of $O(\sigma^3)$ on the number of direction sets. However, the actual number of direction sets is much smaller:

Theorem 2.4 *In σ-geometry the number of direction sets is $\Theta(\sigma)$. In addition, the collection of direction sets can be identified in $\Theta(\sigma)$ time (which is optimal).*

As pointed out by Brazil et al. [24], for λ-geometry this theorem follows almost immediately from the upper and lower bound on meeting angles (Theorem 2.2). Consider two adjacent directions and a Steiner configuration $\{s, x_1, x_2, x_3\}$ with an edge sx_1 that uses these two directions. If λ is not a multiple of 3, then since there are only two feasible meeting angles, both edges sx_2 and sx_3 must be straight. Hence for λ not being a multiple of 3, a direction set contains 4 directions only — one direction set for each pair of adjacent directions (see Figure 8, right). For λ being a multiple of 3, both edges sx_2 and sx_3 can be bent while fulfilling the upper and lower bounds on the meeting angles — hence a direction set contains 6 directions if λ is a multiple of 3. Again, only one direction set is possible if we fix a pair of adjacent directions. Since there are 2σ pairs of adjacent directions, the theorem follows. Furthermore, these 2σ direction sets can trivially be constructed in $O(\sigma)$ time.

For general σ-geometry, proving Theorem 2.4 is a bit more involved. The proof uses a geometric characterization of Fermat-Torricelli points originally given by Chakerian and Ghandehari [35] for smooth and strictly convex unit circles: For a Steiner configuration $\{s, x_1, x_2, x_3\}$, if a unit circle is placed with its centre at s and l_1, l_2 and l_3 are the tangents of the unit circle where the rays $s \rightarrow x_1$, $s \rightarrow x_2$ and $s \rightarrow x_3$ intersect the unit circle, then l_1, l_2 and l_3 form a triangle whose *centroid* coincides with s. We say that the tangents l_1, l_2 and l_3 have the *centroid-property*.

Du et al. [71] used the centroid-property to prove that if the direction of edge sx_1 is fixed, then the directions of edges sx_2 and sx_3 are uniquely determined (for smooth and strictly convex unit circles). Based on a slightly weaker version of the centroid-property from Martini et al. [144] that holds for *general* unit circles, Brazil and Zachariasen [30] showed that if two adjacent directions are fixed — corresponding to the case where edge sx_1 is bent — then edge sx_2 (and edge sx_3) can use at most two fixed adjacent directions. Hence the first part of Theorem 2.4 follows. (Note that direction sets with 5 directions exist in σ-geometry.)

Based on the centroid-property, Pagh [157] gave a $O(\sigma^2)$ time algorithm to construct all direction sets in σ-geometry. The running time was improved to $\Theta(\sigma)$ by Brazil and Zachariasen [30].

Our interest in direction sets stems from the fact that all Steiner points in a full Steiner tree (FST) use the *same* direction set; more precisely, we have the following theorem [30]:

Theorem 2.5 *Given a fulsome FST in σ-geometry, there exists a* single *direction set that is used by* every *Steiner point in the tree.*

(Recall that an SMT is fulsome if it contains the maximum possible number of full components for any SMT on the same terminal set.) For smooth unit circles, including the Euclidean metric, it is known that the edges of a full SMT only use *three* different orientations [71, 139]. Theorem 2.5 can be considered as a generalization of this result to fixed orientation metrics, and it turns out to be crucial in the design of efficient algorithms to construct FSTs (Section 3.1).

Zero-shifts and one bent edge property

The fixed orientation metric is not strictly convex, so in general there are infinitely many SMTs for a given set of terminals. In this section we survey properties pertaining to fulsome FSTs that are part of some SMT.

A *zero-shift* in an FST T is a perturbation of one or more Steiner points such that the perturbation does not increase the length of T. Zero-shifts were introduced by Du and Hwang [73] for $\lambda = 3$, and originally used as a technical tool in the quest for better bounds on the size of the generalized Hanan grid [115, 122, 123], see also Section 2.3. One of the by-products of this work was the following:

Theorem 2.6 *In σ-geometry, every fulsome FST can be embedded in such a way that the FST has at most one bent edge.*

This theorem was proved for $\lambda = 3$ by Du and Hwang [73] in 1992, for $\lambda = 4$ by Lin and Xue [132, 135] in 1998, and finally for arbitrary λ by Brazil et al. [24] in 2000; the full generalization to σ-geometry was given by Brazil and Zachariasen [30] in 2009.

The general technique to prove Theorem 2.6 is to show that if there exist two bent edges e_1 and e_2 in an FST, then we can perturb the Steiner points on the path connecting e_1 and e_2 in such a way that either e_1 or e_2 becomes straight (and without introducing additional bent edges). In fact, we have the following theorem:

Theorem 2.7 *Let e_1 and e_2 be two edges in a fulsome FST T in σ-geometry such that e_1 has an exclusively secondary component and e_2 has an exclusively primary component. Then there exists a zero-shift acting on the Steiner points on the path from e_1 to e_2 in T, such that the shift can continue to be performed until either e_1 has no exclusively secondary component or e_2 has no exclusively primary component. Furthermore, this shift preserves the direction of all straight edges except (possibly) e_1 and e_2.*

Zero-shifts can also be used to prove that FSTs with other special properties exist. As an example, for $\lambda = 3$ or $\lambda = 6$ it is always possible to transform an FST using zero-shifts to another FST where every Steiner point has degree 3 [129]. More elaborate characterizations are presented and utilized in Sections 3.1 and 3.2.

A *fundamental zero-shift* is zero-shift that cannot be decomposed into two zero-shifts each of which acts on a subset of the Steiner points acted on by the fundamental zero-shift (and at least one of which acts on a proper subset of those Steiner points). Any zero-shift can be decomposed into fundamental zero-shifts. Brazil et al. [28, 29] showed that in λ-geometry and for λ being a multiple of 3, zero-shifts perturb *single* Steiner points; for λ not being a multiple of 3, fundamental zero-shifts perturb *two* neighbouring Steiner points. In general σ-geometry, the type of fundamental zero-shift depends on the number of directions in the underlying direction set: For size 4 direction sets 1-point perturbations do not exist, while for size 5 and 6 direction sets 1-point perturbations do exist [30].

Steiner ratio

Consider interconnecting a set of terminals N under a metric given by an arbitrary unit circle $\mathcal{C}$ *without* being allowed to use Steiner points. This corresponds to computing a *minimum spanning tree (MST)* for N — a problem that is polynomially solvable (see Section 1.3).

Let $|SMT_{\mathcal{C}}(N)|$ and $|MST_{\mathcal{C}}(N)|$ denote the length of an SMT and an MST, respectively, for N under the metric given by $\mathcal{C}$. Clearly $|SMT_{\mathcal{C}}(N)| \leq |MST_{\mathcal{C}}(N)|$ since an SMT is a shortest possible interconnection of N. Now the question is: How much shorter can an SMT be relative to an MST for the same set of terminals? Define

$$\rho_{\mathcal{C}}(N) = \frac{|SMT_{\mathcal{C}}(N)|}{|MST_{\mathcal{C}}(N)|}$$

to be the ratio between the length of an SMT and an MST for N. The *Steiner ratio* $\rho_{\mathcal{C}}$ for the metric given by unit circle $\mathcal{C}$ is defined as

$$\rho_{\mathcal{C}} = \inf_{N} \rho_{\mathcal{C}}(N)$$

That is, the Steiner ratio is the smallest possible ratio between the SMT and MST lengths for any set of terminals. In this section we briefly survey the results on the Steiner ratio for λ-geometry and σ-geometry.

Let ρ_λ be the Steiner ratio in λ-geometry. Consider the rectilinear plane ($\lambda = 2$). For the set of terminals $\{(-1,0),(0,-1),(1,0),(0,1)\}$, the SMT is a cross of length 4, while an MST has length 6 (all terminals are separated by a distance of at least 2). Thus we have $\rho_2 \leq 4/6 = 2/3$. Furthermore, as shown in the seminal paper by Hwang [97], we also have $\rho_2 \geq 2/3$, so for $\lambda = 2$ the Steiner ratio problem is fully solved.

For the general Steiner ratio problem in λ-geometry, Sarrafzadeh and Wong [180] gave the inequality

$$\rho_\lambda \geq \frac{\sqrt{3}}{2} \cos\frac{\pi}{2\lambda}$$

which follows from

$$\begin{aligned} |SMT_\lambda(N)| &\geq |SMT_\infty(N)| && (1)\\ &\geq \frac{\sqrt{3}}{2} |MST_\infty(N)| && (2)\\ &\geq \frac{\sqrt{3}}{2} \cos\frac{\pi}{2\lambda} |MST_\lambda(N)| && (3) \end{aligned}$$

Here $|SMT_\infty(N)|$ denotes the length of an SMT for N as $\lambda \to \infty$, which correspond to the Euclidean metric; the second inequality follows from the Steiner ratio theorem for the Euclidean metric [72]. It follows from this lower bound that $\rho_3 \geq 3/4$; a matching upper bound can be obtained by placing three terminals on every second vertex of the regular hexagon (which is the unit circle for $\lambda = 3$).

The exact Steiner ratio is not known for all values of λ (see Open Problem 1 on page 70). In Table 1 we summarize the results known for λ-geometry. It is interesting to note that the Steiner ratio is not a monotonically increasing function of λ. However, both the lower and upper bound approach $\sqrt{3}/2$ as $\lambda \to \infty$ as could be expected.

Metric	Steiner ratio	References
$\lambda = 2$ (rectilinear)	$\frac{2}{3}$	[97]
$\lambda = 3$ (hexagonal)	$\frac{3}{4}$	[122]
$\lambda = 4$ (octilinear)	$\frac{2+\sqrt{2}}{4}$	[184]
$\lambda \equiv 3 \bmod 6$	$\frac{\sqrt{3}}{2} \cos \frac{\pi}{2\lambda}$	[122]
$\lambda \equiv 0 \bmod 6$	$\frac{\sqrt{3}}{2}$	[122]
General λ (lower bound)	$\max\{\frac{2}{3}, \frac{\sqrt{3}}{2} \cos \frac{\pi}{2\lambda}\}$	[71, 180]
General λ (upper bound)	$\min\{\frac{\sqrt{13}-1}{3}, \frac{\sqrt{3}}{2} \frac{1}{\cos \frac{\pi}{2\lambda}}\}$	[58, 71, 180]
$\lambda \to \infty$ (Euclidean)	$\frac{\sqrt{3}}{2}$	[72]

Table 1: Overview of the results on the Steiner ratio ρ_λ for λ-geometry.

For general σ-geometry the bounds on the Steiner ratio are the same as those for arbitrary normed planes, since any norm can be approximated arbitrarily closely by a weighted fixed orientation metric. In normed planes $2/3$ is a tight lower bound on the Steiner ratio [79] (the bound is achieved in, e.g., the rectilinear plane). The best known upper bound is $(\sqrt{13}-1)/3 \approx 0.8686$ [71], but it is conjectured that the (tight) upper bound is $\sqrt{3}/2 \approx 0.8660$ — the Steiner ratio for the Euclidean metric [72].

Results on the Steiner ratio for other metric spaces are surveyed by Cieslik [55, 57]. For L_p metrics it is known that the Steiner ratio is in the interval $[2/3, \sqrt{3}/2]$, where the lower bound is attained for $p = 1$ and $p = \infty$, while the upper bound

is attained for the Euclidean metric L_2 [139]. That the L_1 and L_∞ metrics have identical Steiner ratios follows from the fact that affinely equivalent unit circles achieve the same Steiner ratio; also, if two unit circles have almost the same shape, then their Steiner ratios have almost the same value [58]. Intuitively, this is the reason why the Steiner ratio for $\lambda \to \infty$ approaches the Steiner ratio for the Euclidean metric.

2.3 Generalized Hanan grid

For the rectilinear Steiner tree problem it is known that there exists an SMT in the Hanan grid (see Section 1.2). Sarrafzadeh and Wong [180] pointed out that when going from two to three orientations in the plane, there exist terminal sets for which every SMT has Steiner points that are *not* in the corresponding Hanan grid.

A natural question is therefore: Does there exist a "small" grid structure in which an SMT for the general fixed orientation problem always can be found? As for the rectilinear problem, the existence of such a grid — which induces a weighted planar graph — would make it possible to reduce the fixed orientation problem to the Steiner tree problem in graphs. In this section we survey the results pertaining to this reduction.

For any set of points P, define $\mathbf{GG}(P)$ to be the set of intersection points obtained by drawing lines in all legal orientations through every point in P. Define the *generalized Hanan grid* $\mathbf{GG}_i$ as follows: For terminal set N, $\mathbf{GG}_0(N) = N$ and for $i > 0$ recursively define $\mathbf{GG}_i(N) = \mathbf{GG}(\mathbf{GG}_{i-1}(N))$ (Figure 9). Note that $\mathbf{GG}_1(N)$ coincides with the vertices of Hanan grid for the rectilinear metric.

Theorem 2.8 *For each set of n terminals N there exists an SMT for N such that all Steiner points are in $\mathbf{GG}_{n-2}(N)$.*

This theorem has a fairly long history, and over time it has been shown to hold for increasingly larger classes of fixed orientation problems. The generalized Hanan grid was introduced in 1992 by Du and Hwang [73], and they proved that Theorem 2.8 holds for $\lambda = 3$ (uniform metric with three orientations). In fact, they proved that for $\lambda = 3$, one can perform zero-shifts (by moving one Steiner point at a time) until each FST contains at most one bent edge. The theorem then easily

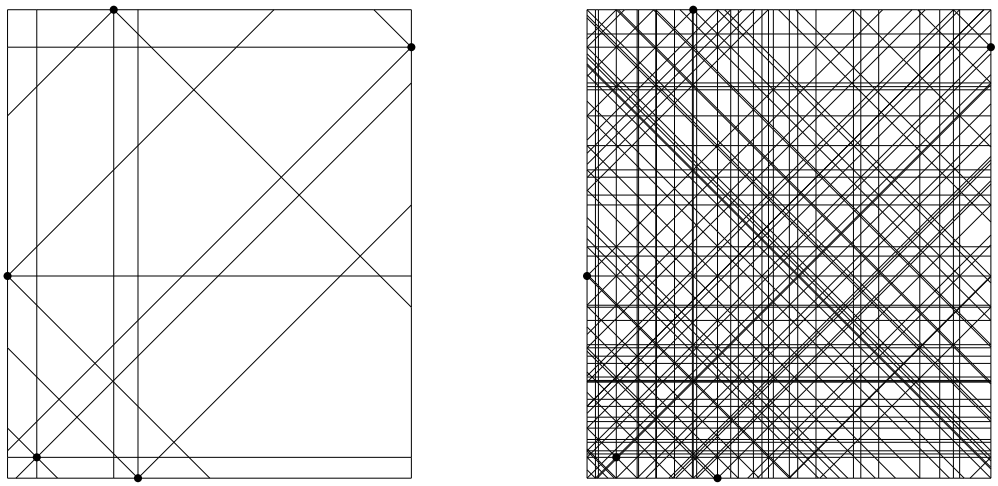


Figure 9: Generalized Hanan grid $\mathbf{GG}_1(N)$ (left) and $\mathbf{GG}_2(N)$ (right) for $\lambda = 4$, and a set N with 5 terminals. Only lines segments that are within the bounding box of the terminals are drawn.

follows since in such an FST there must exist a Steiner point s that is connected with straight edges to *two* terminals; hence s is in $\mathbf{GG}_1(N)$. By removing the two terminals and their straight edges, and considering s to be a (pseudo) terminal, the argument can be repeated for all (up to) $n-2$ Steiner points.

Du and Hwang also conjectured that for any $i > 0$ there exists a fixed orientation metric and a terminal set N such that all SMTs for N have some Steiner point *not* in $\mathbf{GG}_i(N)$. In other words, they conjectured that the bound $n-2$ in Theorem 2.8 cannot be reduced to a constant.

The journey towards proving Theorem 2.8 was as follows. In 1995 Koh [115] and Lee et al. [123] independently proved that the theorem holds for $\lambda = 4$ by showing that it is always possible to perform zero-shifts such that some Steiner point is connected to two terminals using straight edges only. In 1996 Lee and Shen [122] generalized the result to any λ (or uniform orientation metric) using the same proof technique. Finally, in 2001 Li et al. [130] showed that Theorem 2.8 holds for all unweighted fixed orientation metrics, and in 2009 Brazil and Zachariasen [30] proved that the theorem also holds for the weighted case — and hence for any metric given by a polygonal unit circle. The result is not stated explicitly in [30],

but the theorem follows from the fact that there always exists an SMT for which every FSTs has at most one bent edge (Theorem 2.6).

The bound provided by Theorem 2.8 can be improved for special cases of the fixed orientation problem. For $\lambda = 3$ (three uniform orientations) a bound of $\lceil (n-2)/2 \rceil$ follows from the fact that for $n \geq 4$ it is possible to perform zero-shifts in such a way that *two* Steiner points simultaneously are connected to two terminals using straight edges only [225]. For three arbitrary orientations ($\sigma = 3$) only a weaker bound of $\lceil (n-1)/2 \rceil$ is known [130, 131], and for four uniform orientations ($\lambda = 4$), the best known bound is $\lceil 2n/3 \rceil - 1$ [132, 135]. The bounds for $\lambda = 3$ and $\sigma = 3$ are known to be tight; furthermore, it is known that the bound for $\lambda = 4$ must be strictly greater than $\lceil (n-2)/2 \rceil$ [130, 131].

It has been conjectured that the bound in Theorem 2.8 can be improved [22, 132, 135]. There are nevertheless arguments that support the opposite fact, and we conjecture that the bound in Theorem 2.8 is tight (Open Problem 2).

From an algorithmic point of view, Theorem 2.8 has limited use since the number of vertices in $\mathbf{GG}_i(N)$ is $\Omega(n^{i+1}\sigma^i)$ for $|N| = n$, which is $\Omega(n^{n-1}\sigma^{n-2})$ for $i = n-2$. Even for small problem instances, the generalized Hanan grid is too large to be useful (see Figure 9).

2.4 Conclusions

The theoretical results that were presented in this section have played a central role in the development of efficient algorithms for the Steiner tree problem in fixed orientation metrics. It is, however, interesting to note that while early contributions focused on proving results related to the generalized Hanan grid, recent contributions have focused more on proving structural results independent of their Hanan grid applicability, e.g., results related to the distribution of orientations in full Steiner trees. The exact value of the Steiner ratio for all uniform orientation metrics is still among the future challenges — as well as further generalizations of the results to unit circles that have a mixture of polygonal and strictly convex boundary. From a practical point of view, however, the Steiner ratio is less important due to significant progress in approximation and exact algorithms over the last decade.

3 Algorithms

In this section we survey the range of algorithms that have been proposed for computing Steiner minimum trees (SMTs) under uniform and fixed orientation metrics. The survey does not cover all algorithms that have been specifically devised for the *rectilinear* Steiner tree problem; rather, we focus on algorithms that cover *other* uniform and fixed orientation metrics. In most cases the algorithms described work under all fixed orientation metrics.

In the first section we consider the problem of constructing an SMT for a given topology, that is, when the interconnection pattern of terminals and Steiner points is known. We then move on to give an algorithm for constructing the so-called flexibility polygon, which is a representation of all SMTs for a given topology. Finally, we present heuristics and exact algorithms for the NP-hard general case (i.e., when the topology is unknown).

Before we embark on the journey of algorithms for the SMT problem, let us consider an elementary problem — namely the one of computing the length of an SMT for *two* terminals p and q. This is obviously the same as computing the distance between p and q. As pointed out by Shen [184], distances under the *uniform* orientation metric (or in λ-geometry) can be computed in constant time, i.e., independent of λ. The problem reduces to identifying the two legal orientations that are immediate preceding and succeeding legal orientations to the orientation of the straight line between p and q. Due to the regularity of the legal orientations in the uniform case, this is easy to do in constant time. However, achieving constant time appears to be more difficult under a general fixed orientation metric (Open Problem 3); a running time of $O(\log \sigma)$ can be achieved by binary search on the sorted set of σ legal orientations. In many applications σ can be assumed to be bounded by a constant, and in this case distances can trivially be computed in constant time.

3.1 Steiner tree problem for a given topology

The problem of constructing an SMT for an arbitrary given topology can be seen as a further generalization of the general Fermat problem (where the topology is a star). In this section we present efficient algorithms for constructing SMTs for a given *full Steiner topology*, i.e., where all terminals are leaves and all Steiner

points are degree 3 interior vertices. We restrict our attention to topologies where Steiner points have degree 3, since degree 4 Steiner points only appear in very restricted and well-defined cases [30, 54, 56, 193]. The restriction to full topologies is no limitation either, as any topology is a union of full topologies.

For a given (full Steiner) topology $\mathcal{T}$, the length $|T|_{\mathcal{C}}$ of a Steiner tree T with topology $\mathcal{T}$, seen as a function of the location of the Steiner points of T, is a convex function (since T lies in a normed plane). However, this convexity is not strict; it may be possible to move Steiner points in an SMT without changing the length of the tree. A Steiner tree T is said to be *locally minimal* if there is no perturbation of its Steiner points which reduces the length of T. Thus, if T is an SMT for a given topology, it is locally minimal — and vice versa. Note that a locally minimal tree is not necessarily an SMT for the terminals that its spans.

The problem of locating Steiner vertices in a tree with a given topology (in a general metric space) was considered in an early paper by Sankoff and Rousseau [176]. They gave dynamic programming algorithms for some special metrics, including Hamming distance and higher-dimensional rectilinear metrics. For a survey on algorithms and complexity results related to the construction of shortest networks under given tree topologies — covering a range of different metrics — consult the survey by Jiang and Wang [108].

Euclidean problem

The classical Melzak-algorithm [145] computes a full Euclidean SMT for a given full Steiner topology $\mathcal{T}$ in $O(2^n)$ time (where n is the number of terminals), or decides that no such tree exists.

The idea of the algorithm is to root $\mathcal{T}$ in one of the terminals r, and identify a pair of terminals z_1 and z_2 with maximal depth that share an (interior) Steiner point s; such a pair always exists if $n \geq 3$. Consider the oriented line l through z_1 and z_2 (Figure 10). Since the Euclidean metric is convex, the Steiner point s has a unique location in the SMT T for topology $\mathcal{T}$. Assume that s lies on the right hand side of l. Now we may replace z_1 and z_2 by the third point z on the equilateral triangle with z_1z_2 as one of the sides and such that z is on the left hand side of l. The point z is called the *equilateral point* of z_1 and z_2.

Melzak's key contribution was to show that if we solve the new problem with $n-1$ terminals, the Steiner point s can be identified by locating the intersection

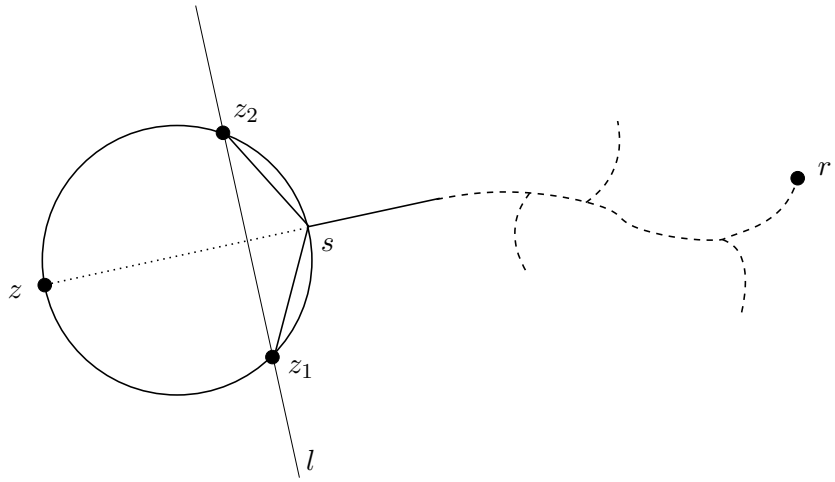


Figure 10: Recursion step of the Melzak-algorithm. Terminals z_1 and z_2 are replaced with the new terminal z and the problem is solved recursively.

between the SMT for the smaller problem with the circle through z_1, z_2 and z (assuming that the SMT for the smaller problem exists). Hence, recursing on the smaller problem eventually results in the trivial problem of constructing an SMT for two terminals; the length of the so-called Simpson line between these two terminals is the length of the SMT for the given topology. The total running time of $O(2^n)$ follows from the fact that in each recursion step, the Steiner point could be on either side of the line through the terminals — and the algorithm has no means of deciding the correct side.

Hwang [99] improved the running time of the Melzak-algorithm to (optimal) linear time $\Theta(n)$ by proposing a clever method to identify the correct side of the Steiner point in each step. Furthermore, this test could be performed in constant time.

The Melzak-algorithm only works if the topology is *non-degenerate*, that is, if no Steiner points overlap with each other or with terminals in the corresponding SMT. For the more general problem, where degenerate SMTs are allowed, Hwang and Weng proposed the so-called "luminary algorithm", which solves the SMT problem for a given topology in $O(n^2)$ time; the average running time is $O(n \log n)$, where the average is taken over all full topologies for the set of terminals [223]. Another general method to solve the degenerate problem is the iterative approach by Smith [188], which also can be generalized to higher dimensions and other

metrics. This method is quite practical, but has the weakness that it only converges with high probability.

In the fixed orientation problem the exact angles around Steiner points are not known in advance (as in the Euclidean problem). As a consequence, there are several candidates for "equilateral" points in each recursion step. As shown by Brazil et al. [24], the Melzak-approach results in an exponential-time algorithm for the fixed orientation problem. Until now it has not been possible to improve the running time using this approach.

Linear programming

The fact that the unit disc $\mathcal{D}$ for the weighted fixed orientation metric is linearly constrained (and convex) makes it possible to compute distances using linear programming — and hence to compute SMTs for (any) given topology by solving a linear programming problem.

In 1999, Thurber and Xue [199] gave a linear programming formulation for the $\lambda = 3$ case, and in 2002, Xue and Thulasiraman [224] generalized the formulation to the general uniform orientation metric. Zachariasen [234] pointed out a nontrivial error in this formulation, and presented a new and correct formulation, which is briefly presented here.

Let $\mathbf{u}_l$, $l = 0, \ldots, 2\sigma - 1$, be the 2σ vectors that define the vertices of the unit circle $\mathcal{C} = \partial\mathcal{D}$ (in counter-clockwise order around the circle). The successor of unit vector $\mathbf{u}_l$ is the vector $\mathbf{u}_{l+1}$, where $l + 1 = 0$ when $l = 2\sigma - 1$. Assume that we would like to compute the distance $|pq|_{\mathcal{C}}$ between two points p and q under the metric given by $\mathcal{C}$. Let $\{\alpha_l, \beta_l\}$ be the unique solution to

$$q = p + \alpha_l \mathbf{u}_l + \beta_l \mathbf{u}_{l+1} \tag{4}$$

for each $l = 0, \ldots, 2\sigma - 1$. Using fairly simple arguments, Zachariasen [234] showed that

$$|pq|_{\mathcal{C}} = \max_{l \in \{0,\ldots,2\sigma-1\}} (\alpha_l + \beta_l) \tag{5}$$

and therefore that $d_{pq} = |pq|_{\mathcal{C}}$ can be computed by solving the following linear program:

$$\begin{array}{ll} \text{minimize} & d_{pq} \\ \text{subject to} & \alpha_l + \beta_l \;\leq\; d_{pq}, \quad l \in \{0, \ldots, 2\sigma - 1\} \end{array}$$

Note that α_l and β_l depend linearly on the coordinates of p and q. By applying the constraints in the above formulation for each of the edges in the given topology, an SMT for this topology can be computed in polynomial time (in the size of the input). Clearly, degenerate and non-tree topologies can also be handled by this formulation.

A special case of the above problem, the (separable) rectilinear problem [32], is known to be the dual of a transshipment problem. Similar connections are currently unknown for the general problem.

Linear time algorithm for general σ-geometry

In order to compute an SMT for a given full Steiner topology in linear time, it is necessary to use an algorithmic approach that is different from the Melzak-algorithm. In this section we present the historical time-line and some of the technical results leading up to the following theorem.

Theorem 3.1 *Given a full Steiner topology $\mathcal{T}$ with n terminals and a weighted fixed orientation metric with σ legal orientations, then in $O(\sigma n)$ time we can either construct a full and fulsome SMT T with topology $\mathcal{T}$, or decide that no such tree exists.*

Hwang [97] proved the theorem for the rectilinear metric ($\lambda = 2$) in 1976. Hwang showed that there always exists a rectilinear SMT for which the full Steiner trees take on restricted canonical forms — so-called Hwang topologies — where all terminals are connected to a backbone in a chain-like fashion. Computing an SMT for such a topology can trivially be done in linear time.

In 1997, Li et al. [127, 128] gave an interesting structural result for the hexagonal metric ($\lambda = 3$). They showed that there always exists a full SMT with a single bent edge and such that every Steiner point has edges making 120° with each other. Albeit not pointed out by Li et al., this immediately gives an $O(n^2)$ time algorithm for computing an SMT for a given full Steiner topology: Choose a candidate for the bent edge (there are $O(n)$ such edges), root the topology in this edge and use a Melzak-approach to construct each of the subtrees in a bottom-up fashion. By using a more clever implementation, similar to the one described below, linear running time can be obtained.

For the case where λ is a multiple of 3, Nielsen et al. [155] in 2002 obtained a linear time algorithm for constructing an SMT for a given full Steiner topology by showing that there always exists an SMT where every Steiner point coincides with the Steiner point for the corresponding *Euclidean* SMT; hence by using the Melzak-Hwang algorithm to locate the Steiner points, linear running time can be achieved.

A proof of Theorem 3.1 for all *uniform* orientation metrics was given by Brazil et al. [27] in 2006. The concept of "forbidden subpaths" [25, 26] played a crucial role in achieving this fast algorithm. Brazil and Zachariasen [30] proved Theorem 3.1 in full generality by using the same algorithmic approach, but by simplifying several of the underlying technical results — and without reference to the results related to forbidden subpaths. In the remainder of this subsection, we sketch some of the key ideas of this algorithm.

As a consequence of Theorem 2.7, it follows that there always exists a so-called *canonical SMT* defined as follows:

Definition 3.1 *Given an ordering of the edges in a full Steiner topology $\mathcal{T}$, an SMT T for topology $\mathcal{T}$ is said to be* canonical *with respect to that ordering if T contains at most one bent edge and all primary edges of T come before secondary edges of T under the ordering.*

Hence, in a canonical tree there exists a so-called *transition edge* (which is possibly a bent edge), such that all edges that appear before the transition edge wrt. the given ordering are *primary* (straight) edges and all edges that appear after the transition edge are *secondary* (straight) edges.

The concept of canonical SMTs forms the cornerstone of the $O(\sigma n)$ algorithm for constructing a full and fulsome SMT for given topology. More specifically, the algorithm attempts to construct an SMT for each direction set (of which there are $\Theta(\sigma)$, see Theorem 2.4). Given a direction set, the SMT — if it exists — is then constructed in $O(n)$ time. Here we first outline a simple $O(n^2)$ algorithm for this problem, and then we briefly explain how $O(n)$ running time is obtained.

Consider any fixed ordering of the edges of the topology $\mathcal{T}$. Now, assuming that a full and fulsome SMT exists, there must also exist a canonical tree T under the given ordering. Assume that edge number k under the ordering is the transition edge. Then all edges numbered lower than k are primary edges and all edges

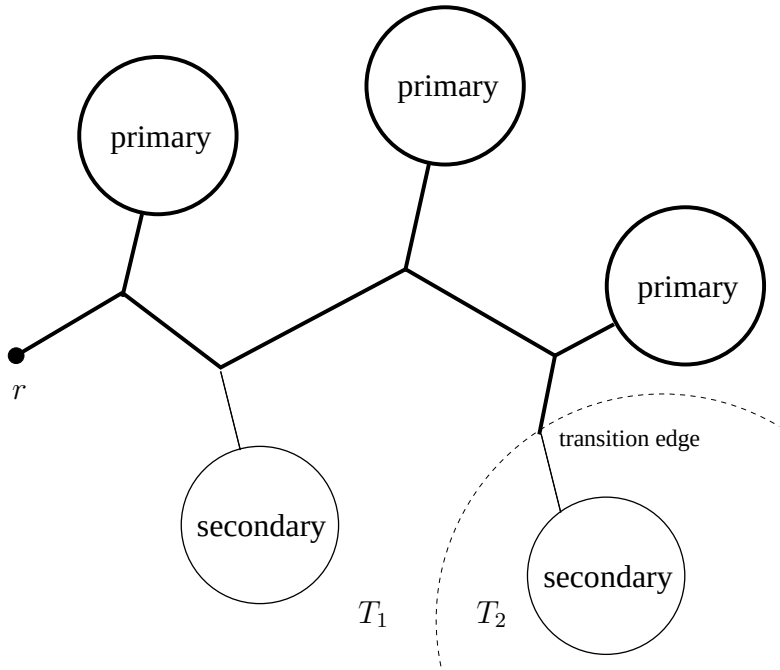


Figure 11: Illustration of a tree T with the depth-first order canonical form rooted at r. Primary edges are drawn as bold edges, and the remaining edges are secondary edges.

numbered higher than k are secondary edges. It turns out that this information is sufficient to make it possible to construct each of two subtrees relative to the transition edge in a bottom-up manner in $O(n)$ time [27, 30]. Enumerating over all choices for the transition edge results in a $O(n^2)$ algorithm.

Turning the quadratic time algorithm into a linear-time algorithm requires that we carefully choose an ordering of the edges. A depth-first ordering of the edges from a given (terminal) root results in a powerful and useful canonical form (Figure 11). By applying appropriate preprocessing, each transition edge can be tested in constant time, resulting in a linear-time algorithm for constructing a full and fulsome SMT for a given topology.

3.2 Flexibility polygon

SMTs in fixed orientation metrics are usually not unique. Non-unique SMTs are *flexible* in the sense that we may choose among several (lengthwise equally good)

embeddings of these SMTs. Hence, flexibility is a measure of the extent to which edges and Steiner points in the minimum length network can be perturbed without increasing the length of the network. This has important applications in solving multi-objective optimisation problems in chip design, involving minimization of negative effects of properties such as congestion or signal delay as a secondary objective [18, 19, 163].

In this section we characterize flexibility formally by defining the *flexibility polygon* for a given topology (and for each of the Steiner points in this topology). This concept was introduced by Brazil, Winter and Zachariasen [28, 29], and they furthermore gave an efficient algorithm to construct the flexibility polygon. The original algorithm [29] was only given for λ-geometry, but it can easily be generalized to arbitrary weighted fixed orientation metrics by applying the structural results from Brazil and Zachariasen [30].

The results related to the flexibility polygon subsume some of the earlier work on flexibility. For the $\lambda = 3$ case, and considering the problem of constructing an SMT with three terminals and one Steiner point, Yan et al. [225] showed that the set of feasible Steiner points forms a region bounded by an equilateral triangle (Figure 12). Li et al. [128] gave a simple algorithm to construct this triangular region based on finding median points and so-called mid-orientation lines. More generally, Shen [184] and Hayase [90] independently showed that when λ is a multiple of 3, the feasible region (called a "public domain" in [184] and a "diamond area" in [90]) is a convex polygon with up to six vertices; when λ is not a multiple of 3, then the SMT for three terminals is unique and the feasible region contains a single point.

The fact that the feasible region for Steiner points is a convex polygon is not surprising given that the problem of constructing an SMT for a given topology can be solved by linear programming, where the coordinates of Steiner points are variables [224, 234]. Even if linear programming in principle can be used to construct feasible Steiner point regions, this would not be as efficient as direct computation as shown below.

For a set of terminals N and a full Steiner topology $\mathcal{T}$ for N, we denote by $S(N, \mathcal{T})$ the *set* of all full and fulsome SMTs interconnecting N with topology $\mathcal{T}$. The *flexibility polygon* for a terminal set N and a full Steiner topology $\mathcal{T}$ is defined to be the *union* of the embeddings of all SMTs in $S(N, \mathcal{T})$. It turns out that this union forms a simply connected region with a polygonal boundary whose vertices include the terminals N [29]. Some examples of flexibility polygons for

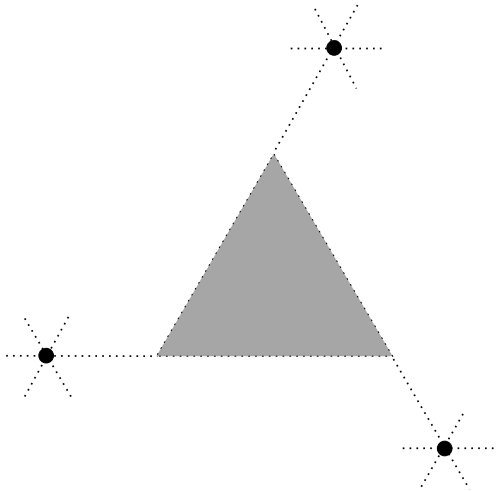


Figure 12: Triangular flexibility polygon for three terminals and $\lambda = 3$. The Steiner point may be placed anywhere in the gray-shaded region.

the cases where $\lambda = 4$ and $\lambda = 6$ are shown in Figure 13.

In the remainder of this section we briefly discuss the following theorem by Brazil, Winter and Zachariasen [29]:

Theorem 3.2 *Given a full Steiner topology $\mathcal{T}$ with n terminals and a weighted fixed orientation metric with σ legal orientations, the flexibility polygon for $\mathcal{T}$ can be computed in $O(\sigma n)$ time.*

The first step of the algorithm is to compute an SMT T by applying Theorem 3.1. The SMT T implicitly identifies a direction set that is used by every Steiner point in T. Recall that a direction set consists of three sets of directions. Each set contains either one or two directions; in the latter case one primary and one secondary direction, and in the former case one direction that can be thought of as being both primary and secondary (see Section 2.2). It should be noted that SMTs that use a direction set with 5 or 6 directions usually have much more flexibility than SMTs that use a direction set with 4 directions. (As an example, SMTs in λ-geometry where λ is a multiple of 3 usually have more flexibility since for these SMTs the corresponding direction sets have 6 directions.)

Consider a counter-clockwise outer walk of T, beginning and ending at the same terminal. This defines a set of *concave paths* in T that have terminals as endpoints

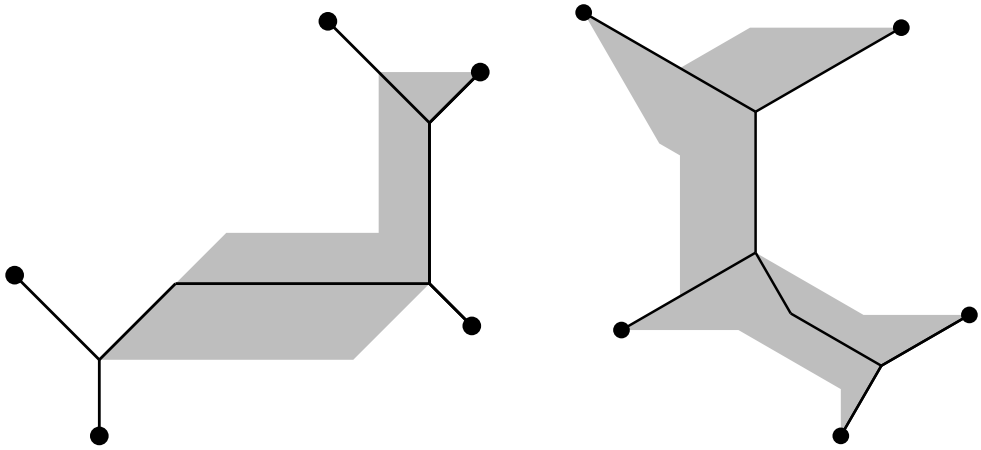


Figure 13: Examples of λ-SMTs and flexibility polygons. Notice that a flexibility polygon may have overlapping boundary segments, indicating that parts of some edges may exhibit no flexibility at all.

and Steiner points as interior points. In other words, these are paths between terminals where at each intermediate Steiner point the rightmost outgoing edge is taken.

For each concave path we now seek an embedding of the SMT that pushes the path as far as possible to the right, defining a so-called *rightmost* concave path. This means that for each edge e on the rightmost concave path there is no alternative embedding where the same edge is to the right of e. It can be shown that the collection of these rightmost concave paths defines the flexibility polygon.

Consider a concave path $P = v_1 v_2 \dots v_{k-1} v_k$ connecting two terminals v_1 and v_k. We define an ordering of the edges of $\mathcal{T}$ by making a depth-first traversal from v_1. At every Steiner point v_i, the subtree of $\mathcal{T}$ rooted at v_i (and not intersecting P) is traversed before the edge $v_i v_{i+1}$ is traversed. The main technical result is now that the SMT that has the canonical form given by this ordering defines the embedding of the rightmost concave path from v_1 to v_k [29].

In order to compute these rightmost concave paths efficiently the algorithm first constructs all primary and secondary subtrees of $\mathcal{T}$, that is, embeddings that consist of primary resp. secondary directions only. This can be accomplished in $O(n)$ time even though there are $O(n)$ potential subtrees. The algorithm works bottom-

up by maintaining a queue of subtrees that can be constructed at a given point in time.

The final construction of rightmost concave paths, and hence the flexibility polygon, is achieved essentially by traversing a primary subtree as far as possible along a concave path — and then switching to a secondary subtree rooted at the opposite end of the edge where the primary subtree ends. The construction only requires one outer walk of T and constant time for each edge traversed — in total $O(n)$ time.

For a given Steiner point s in $\mathcal{T}$, the union of all feasible positions of s in the SMTs in $S(N, \mathcal{T})$ is denoted the flexibility polygon for Steiner point s. Given the flexibility polygon for $\mathcal{T}$, the flexibility polygon for s can be constructed in constant time and has at most six vertices [29]. Examples of such flexibility polygons are shown in Figure 14.

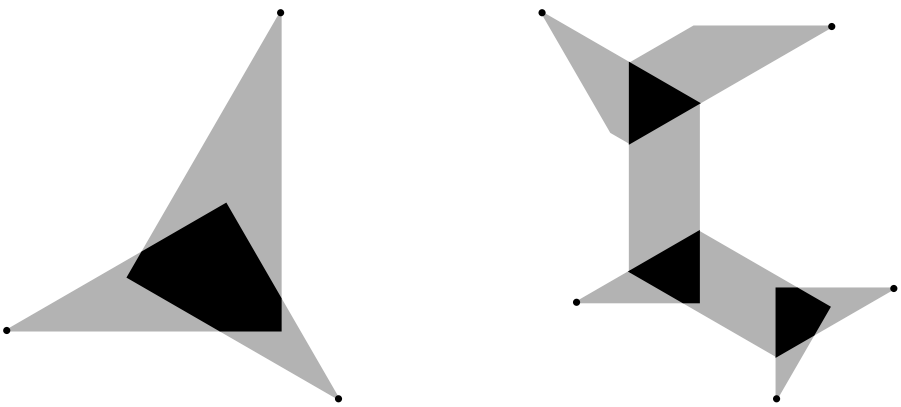


Figure 14: Examples of flexibility polygons (dark-shaded) for Steiner points ($\lambda = 6$). The flexibility polygon on the left has 6 vertices, while the flexibility polygons on the right have 3, 4 and 5 vertices, respectively.

3.3 Heuristics for the general case

In this section we give a survey on heuristics for the general SMT problem: Given a set N of n terminals, construct a shortest interconnection of the terminals under a fixed orientation metric (thus the topology of the tree is unknown). Recall that this

is an NP-hard problem — therefore heuristics that quickly construct near-optimal solutions are valuable in an application setting.

From a theoretical point of view, SMTs can be approximated arbitrarily closely. The polynomial-time approximation scheme (PTAS) of Arora [5] works for any fixed orientation metric, and therefore it is possible to construct a solution to the SMT problem that is within a factor of $1 + \epsilon$ from optimality in polynomial time for any fixed $\epsilon > 0$. However, the degree of the polynomial for small values of ϵ is too large to make this algorithm practical.

Almost all practical alternatives to the Arora algorithm use a minimum spanning tree (MST) for the terminals as a starting point. The reasons are twofold. Firstly, an MST for n terminals can be computed efficiently in $O(n \log n)$ time under any fixed orientation metric [215] (see also Section 1.3). Secondly, as a consequence of the known lower bounds on the Steiner ratio (see Section 2.2), MSTs are provably good approximations to the Steiner tree problem; an MST is at most 50% longer than an SMT under any fixed orientation metric [71, 79, 180]. Furthermore, in practice MSTs are significantly better approximations than their corresponding (worst-case) Steiner ratios indicate. Hence starting with an MST and iteratively performing operations on the tree that reduce the length of tree is an obvious choice.

Since all the surveyed heuristics have the same theoretical approximation guarantee — namely the guarantee provided by an MST — we instead compare their experimental behaviour. In Table 2 we present the average reduction over the MST length for uniformly distributed terminal sets of size around 50 (as reported by the authors). It should be noted that the table does not report the running times of the heuristics which varies significantly.

Optimal embedding of MST

Consider an MST T for a set of terminals N. Recall that each edge pq of T can be embedded (or drawn) in the plane using at most two adjacent legal orientations; any (monotone) zigzag-path consisting of line segments having these two legal orientations is a shortest path under the fixed orientation metric (see Section 2.1). For a given embedding of T, line segments from different edges of T may overlap; if we drop all but one of the overlapping line segments, we obtain a Steiner tree for N. An *optimal embedding* is one where the resulting Steiner tree has the shortest

Algorithm	$\lambda = 3$	$\lambda = 4$
Optimal embedding of MSTs [95, 136]	2.5% [136]	-
Delaunay-based heuristic [187]	-	4.0% [184]
Iterated 1-Steiner heuristic [112]	3.4% [137]	4.1% [115]
Edge insertion heuristic [36]	3.4% [129]	4.1% [129]
Edge-based heuristic [14]	-	4.3% [111]
Greedy triple contraction [238]	-	4.4% [111]
Hierarchical construction [179]	-	1.8% [184]
Simulated annealing based heuristic [128]	3.1% [128]	-
Steiner tree conversion [46]	-	3.6% [46]
SMT	4.1% [154]	4.5% [154]

Table 2: Overview of results for practical heuristics for computing hexagonal and octilinear Steiner trees. The table presents average length reductions over the corresponding MST for uniformly distributed terminal sets of size around 50. For each algorithm there is a reference to the orignal paper describing the algorithm (or algorithmic idea).

possible length — or where the total overlap is as large as possible (Figure 15).

The first heuristic designated for general λ-geometry was based on optimal MST embedding. In 1991, Burman et al. [33] presented a polynomial-time algorithm to compute an optimal embedding for $\lambda = 4$. Their algorithm is a generalization of a similar algorithm for the rectilinear problem [95]. Here we present the main ideas of the original algorithm.

Consider an MST T. We say that T is *separable* if only *adjacent* edges pq and pq' in T can possibly be embedded with overlap. Ho et al. [95] proved that there

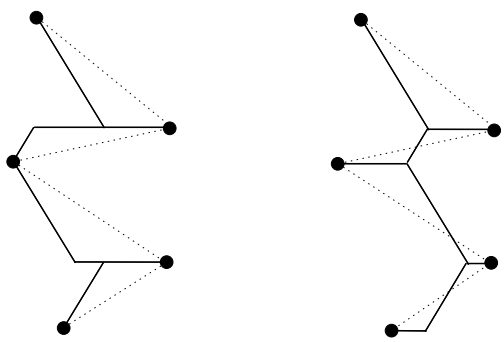


Figure 15: Examples of embedding of an MST for $\lambda = 3$. The topology of the MST is given by the dotted lines. The embedding on the right is shorter than the embedding on the left.

always exists a separable MST for the rectilinear problem ($\lambda = 2$), and furthermore, that such an MST can be computed in $O(n \log n)$ time. This fact makes it possible to devise an efficient algorithm to compute an optimal embedding of a rectilinear MST.

A *Z-shaped* embedding is one where each edge is drawn using at most three line segments (note that the embeddings in Figure 15 are Z-shaped). It turns out that there always exists an optimal embedding of an MST which is Z-shaped. Ho et al. [95] gave a $O(n^7)$ algorithm to construct an optimal Z-shaped embedding — and hence an optimal embedding — for the rectilinear problem. Using similar techniques, Lin et al. [136] gave a $O(n^2)$ time algorithm to construct an optimal embedding of an MST for $\lambda = 3$; an improved $O(n)$ time algorithm was given by Lin and Xue [134]. We conjecture that it is possible to construct a separable MST in $O(n \log n)$ time and to construct an optimal embedding in $O(n)$ time under any fixed orientation metric (Open Problems 4 and 5).

Delauney triangulation based heuristics

In 1981, Smith, Lee and Liebman [187] proposed a fast $O(n \log n)$ heuristic for the Euclidean problem. This heuristic uses the (Euclidean) Delaunay triangulation to identify triplets and quartets of terminals for which small SMTs are constructed; these SMTs form building blocks for a heuristic solution. The SMTs are then greedily, as in Kruskal's MST algorithm, combined with MST-edges to form a Steiner tree.

Lee et al. [123] implemented this algorithm for $\lambda = 4$; only the construction of small SMTs was modified to take the new metric into account. They also used the Euclidean Delaunay triangulation to identify triplets and quartets. Shen [184] implemented the algorithm using λ-geometry Delaunay triangulations. Shen also presented a local refinement and a simulated annealing based algorithm where the first greedy solution is iteratively improved by inserting small SMTs. (This is similar to the approach independently proposed by Winter and Zachariasen [230, 236] for the Euclidean problem). Another Delaunay-based heuristic was suggested by Ho et al. [96] as part of a full-chip routing algorithm.

Iterated 1-Steiner heuristics

The idea of the iterated 1-Steiner heuristic is to iteratively identify a Steiner point s such that the length of an MST for $N \cup \{s\}$ is minimized. (The so-called 1-Steiner tree problem was originally considered for the Euclidean problem by Georgakopoulos and Christofides [82].) Kahng and Robins [112] presented several variants of this approach for the rectilinear problem.

The first application to general λ-metrics was given by Koh [115], who used the generalized Hanan grid to identify Steiner point candidates for $\lambda = 4$. A quadratic-time algorithm to solve the 1-Steiner problem for $\lambda = 3$ was given by Lin et al. [137]; they also presented experimental results for the iterated 1-Steiner heuristic for $\lambda = 3$. A related heuristic based on inserting Steiner points into MSTs was given by Hayase [91].

Edge-based heuristics

Instead of inserting Steiner points into the tree, Chao and Hsu [36] suggested an edge-insertion heuristic for the rectilinear problem. The idea is to insert a new edge into the current tree, and delete the longest edge from the loop that is created. Li et al. [129] extended the method to insert more than one edge at at time. Also, Steiner points were relocated to optimal positions after each insertion, so that the tree becomes locally minimal.

Another edge-based heuristic was proposed for the rectilinear problem by Borah et al. [14, 15]. This algorithm is basically a Steiner point insertion heuristic; however, two of the three vertices involved in the insertion should be connected by an edge in the tree. This reduces the number of possible triples; in fact, there are at most $O(n)$ relevant triples and these can be identified in $O(n \log n)$ time. This approach was used by Madden and Koh [116] in their feasibility study of non-Manhatten architectures, but no detailed results for the reductions over MST-length were given. Kahng et al. [110, 111] presented experimental results for the heuristic for $\lambda = 4$. Zhu et al. [239] proposed a variant of the edge-based method where a spanning graph is used to reduce the number of candidates for edges to insert.

Greedy triple contraction heuristics

The greedy triple contraction algorithm of Zelikovsky [238] for the Steiner tree problem in graphs was the first to improve the (trivial) approximation bound provided by the minimum spanning tree. In addition to being theoretically interesting, this algorithm also has practical potential for computing Steiner trees in fixed orientation metrics. Kahng, Mandoiu and Zelikovsky [110, 111] presented an efficient variant of Zelikovsky's original algorithm; experimental results were presented for the rectilinear and octilinear problem. It is straightforward to extend the algorithm to any fixed orientation metric.

The idea of the algorithm is to insert triples (Steiner point plus edges to neighbouring vertices) in a greedy manner, where the longest edges in the two loops created are removed from the tree. The *gain* of a triple is the net effect of the insertion: The edge length of the removed edges minus the edge length of the triple. The algorithm iteratively chooses a triple with maximum gain, and contracts it, that is, collapses the three vertices and the Steiner point into a single new terminal. In the variant suggested by Kahng et al., triples are inserted in *batches*, hence reducing the running time significantly. A set of $O(n \log n)$ candidate triples is identified in $O(n \log n)$ time; this set includes the so-called empty triples which are triples that do not have any terminals within their bounding rectangle. Finally, an efficient data structure to identify the longest edge on the path between two terminals is given. The total running time of the algorithm is $O(n \log^2 n)$, which makes it applicable to large-scale problem instances.

Similar to the edge-based heuristic, Zhu et al. [239] proposed a variant of the greedy triple contraction heuristic where a spanning graph is used to reduce the number of candidates for triples to insert. This algorithm constructs slightly shorter trees than the algorithm by Kahng et al., but at the cost of a significant increase in asymptotic running time.

Other heuristics

In this section we briefly describe other heuristic approaches that do no fit into the classes given above.

In the hierarchical construction heuristic [180], a binary decomposition is made from the MST-topology. This decomposition then guides a bottom-up tree con-

struction — which involves adding edges (and embeddings of these edges) that minimize total length. Sarrafzadeh and Wong [180] presented experimental results for the rectilinear problem only, but a general algorithm for the uniformly oriented problem was described and analyzed. The heuristic was implemented for $\lambda = 4$ by Shen [184], who also discussed how small optimal SMTs could be computed using the generalized Hanan grid.

Li et al. [127, 128] suggested a simple simulated annealing heuristic for $\lambda = 3$: Select a random set S of $n - 2$ Steiner points from the Hanan grid (for $\lambda = 3$), and compute an MST for $N \cup S$. Simulated annealing steps are performed by relocating Steiner points to adjacent Hanan grid vertices.

In the octilinear heuristic by Chiang and Chiang [44, 46], the idea is to compute a rectilinear Steiner tree and then convert all rectilinear edges to octilinear edges. Also, Steiner points are moved to optimal locations under the octilinear metric. The topology of the tree is not changed.

Finally, Coulston [68] proposed a heuristic for hexagonally partitioned space; the unweighted case of this problem corresponds to the hexagonal problem ($\lambda = 3$). A genetic algorithm that combines full Steiner trees was implemented for the problem.

Summary of heuristic performance

As indicated by Table 2, the edge-based heuristic and the greedy triple contraction heuristic outperform the other heuristics wrt. quality. Even more noteworthy, these two heuristics also have good running time behaviour (clearly the running time of the optimal embedding heuristic is hard to beat). The edge-based heuristic has quadratic $O(n^2)$ running time, while the greedy triple contraction heuristic has sub-quadratic $O(n \log^2 n)$ running time behaviour.

Given the excellent performance (both wrt. quality and speed) and the ease with which the greedy triple contraction heuristic can be applied to other metrics and more general interconnection problems, this heuristic must be considered to be the "champion" among practical heuristics for the fixed orientation Steiner tree problem.

3.4 Exact algorithms for the general case

Since the fixed orientation Steiner tree problem is NP-hard, there is little hope that there exists a provably efficient exact algorithm. However, for restricted problem instances polynomial-time algorithms do exist. As an example, Lin and Xue [133] devised a linear-time algorithm for the problem of computing an hexagonal SMT when the terminals are distributed on the boundary of a regular hexagon; Lin and Xue also observed experimentally that for $n \geq 10$, all computed SMTs were minimum spanning trees (see Open Problem 6).

In this section we present two *practical* approaches for computing SMTs for general problem instances. These algorithms have no provably good running time bounds, but as we shall see, the so-called Geosteiner approach works very well in practice. Using the Geosteiner approach, it is possible to compute SMTs for problem instances with randomly distributed terminal sets — and real-life problem instances from chip design — with several thousand terminals.

Enumeration of full topologies

The problem of computing a Steiner tree for a given full Steiner topology was considered in Section 3.1. One straightforward approach to solving the general case is to enumerate all full topologies, and for each of these to compute an SMT. (Here we assume that we use an algorithm that also considers degenerate topologies — such as the linear programming approach presented in Section 3.1.)

Enumeration of full topologies was proposed as a solution method for the Euclidean problem by Smith [188], who also gave an elegant representation of full topologies. Consider the unique topology $\mathcal{T}_3$ for three terminals consisting of a Steiner point that is connected to three leaf terminals. When adding a fourth terminal t to this topology, we may view this as "joining" t to one of the edges uv in $\mathcal{T}_3$ by removing uv and connecting a Steiner point s to t, u and v. By generalizing this argument, it is easy to see that a topology $\mathcal{T}_n$ with $n \geq 4$ terminals can be represented by a $n-3$ vector $V(\mathcal{T}_n) = (v_4, v_5, \ldots, v_n)$, where $1 \leq v_i \leq 2i-5$. (This also shows that the number of full Steiner topologies on n terminals is $3 \cdot 5 \cdots (2n-5)$, a function which grows super-exponentially in n.)

Based on an efficient application of this representation, Thurber and Xue [199] in 1999 implemented the first exact algorithm for the hexagonal metric ($\lambda = 3$).

Their branch-and-bound algorithm works as follows. First assume that an upper bound *UB* on the length of an SMT exists — as provided by a heuristic method. Also assume that the terminals are ordered by some ordering scheme. Let $\mathcal{T}_k$ be a topology on the first k terminals ($k < n$), and let $V(\mathcal{T}_k) = (v_4, v_5, \ldots, v_k)$ be the corresponding vector. If an SMT for topology $\mathcal{T}_k$ has length greater than or equal to *UB*, then no matter how $\mathcal{T}_k$ is extended by joining terminals (as given by the ordering), this cannot result in a tree with a shorter length than *UB*. Thus we need not consider any vector for the complete problem with prefix $(v_4, v_5, \ldots, v_k)$, essentially cutting off a whole branch in the search tree.

Using this approach, Thurber and Xue were able to solve the SMT problem for 10 terminals in seconds. Computing an SMT for 20 terminals took several hours, and was not feasible in all cases. The branch-and-bound scheme proved to be very effective; as an example, approximately 2 million full topologies for 10 terminals were reduced to around 350 topologies for which linear programming was applied to compute an SMT. Even though the branch-and-bound approach helps, the running time growth of full topology enumeration makes the approach infeasible even for moderately sized problem instances.

Geosteiner approach

Recall that an SMT is a union of full Steiner trees (FST), which are SMTs where all terminals are leaves and all Steiner points are interior vertices. Instead of enumerating full topologies, it turns out to be much more efficient to enumerate FSTs and then combine these to form an SMT for the complete set of terminals; this is called the *Geosteiner approach* in the following.

The Geosteiner approach has its origins in the work on the Euclidean problem by Cockayne [60] (in the 1970s) and Winter [216] (in the 1980s). More recently, significant improvements were obtained, allowing the solution of Euclidean and rectilinear Steiner tree problem instances with several thousand terminals [89, 206, 207, 208, 209, 220, 231].

Nielsen, Winter and Zachariasen [154] adapted the Geosteiner approach to the uniform orientation (λ-geometry) Steiner tree problem. The efficiency of the approach depends critically on the characterization of canonical forms given by Brazil et al. [27] (see Section 3.1). Here we describe the main ideas of the FST enumeration (or generation) algorithm proposed by Nielsen et al. [154]. It should

be noted that the so-called concatenation phase of the algorithm — the combination of FSTs into a complete SMT — is independent of the underlying metric, and was solved using the branch-and-cut algorithm proposed by Warme [206].

Consider any FST F in λ-geometry. We may assume that all Steiner points of F have degree 3, and that F has at most one bent edge (see Section 2.2). Let c be the corner point of the bent edge. (If F has no bent edges then let c be the middle point of an arbitrary straight edge of F.) If we split F at c, we obtain two so-called *half FSTs* — each of which is an FST with straight edges that has one "dangling" extension ray (Figure 16). The base of the extension ray is called the root of the half FST. Note that any FST can be obtained by joining two half FSTs; the intersection between the extension rays defines the corner point of the single bent edge in the FST. Also, any half FST can be obtained by joining two half FSTs; here the intersection becomes the root of the half FST and a new extension ray is added with the new root as its base (Figure 16).

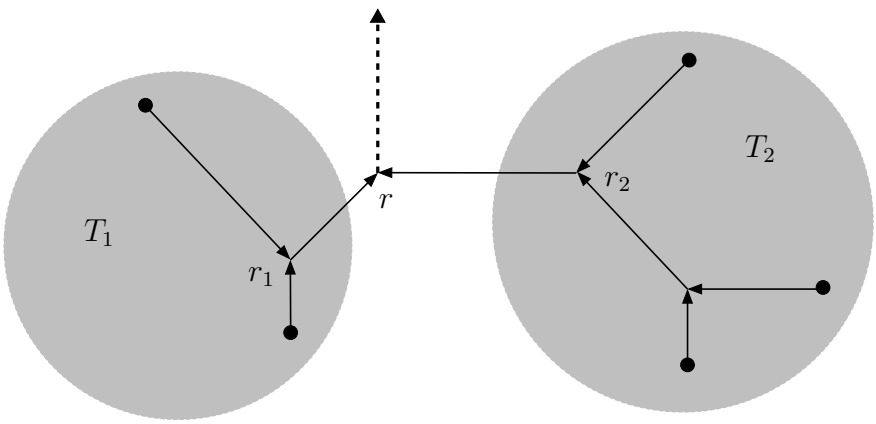


Figure 16: Half FSTs T_1 and T_2 with roots r_1 and r_2 are joined to form a larger half FST with root r. The dashed arrow is the extension ray of the new half FST.

In the FST generation algorithm, half FSTs are generated by increasing size (which is the number of terminals spanned). A single terminal with an extension ray forms a size 1 half FSTs. Clearly there are $2\lambda n$ such half FSTs. A size 2 half FST can be obtained by joining two size 1 half FSTs, and more generally, a size i half FST can be obtained by joining a size j half FST with a size $i - j$ half FST (where j ranges from 1 to $\lfloor i/2 \rfloor$). The power of this idea comes from the *pruning* techniques employed. These are techniques that can eliminate a half FST from consideration since it can be shown that the half FST cannot be part of any SMT.

Elementary pruning is based on for example angle properties of SMTs in λ-geometry (see Section 2.2), and on properties such as the lune property which basically states that terminals cannot be too close to long edges. Bottleneck distance properties — which bound the edge lengths in an SMT by the edge lengths in a corresponding MST — are powerful for λ-geometry (as they are for the Euclidean and rectilinear metric). However, these pruning techniques alone are not sufficient to solve large-scale problem instances.

The concept of canonical forms facilitates pruning based on a cleverly chosen canonical form. The idea is to choose a canonical form that can be tested bottom-up in the algorithm, that is, which has effect even for small half FSTs. This can be achieved by using the edge ordering that comes from a depth-first traversal starting in the lowest indexed terminal of each FST (under any given ordering of the terminals). Why is this canonical form effective? Consider a pair of half FSTs that are to be joined. Either the lowest indexed terminal is among the terminals spanned by the two half FSTs, or it is "outside" (and thus unknown). In the former case, we know which terminal has the lowest index, and this restricts the distribution of directions in the merged half FST. In the latter case, even more restrictions can be enforced; if neither the restrictions in the former nor the latter case can be fulfilled, the resulting half FST can be pruned.

The resulting number of generated FSTs is almost linear in practice, and the size of the largest generated FST is practically bounded by a constant as n increases. Using this algorithm, Nielsen et al. [154] were able to solve randomly generated problem instances with up to 1000 terminals in less than one hour for $\lambda \leq 8$. A single problem instance with 10000 terminals was solved in less than two days for $\lambda = 4$.

Independently of the work of Nielsen et al. [154], Coulston [67] implemented a similar algorithm for $\lambda = 4$. The algorithm could handle problem instances up to around 25 terminals — mainly since some of the more sophisticated pruning techniques were not used, among these pruning tests based on canonical forms. More recently, Pagh [157] adapted the Geosteiner approach to the general weighted fixed orientation metric. Since no pruning tests based on canonical forms were employed, this algorithm did not scale well either.

3.5 Conclusions

The progress in algorithms for solving fixed orientation Steiner tree problems has been enormous over the last decade. For the problem of computing an SMT for a given topology, new algorithmic approaches were needed to solve the problem in linear time. Similar techniques were capable of computing the flexibility polygon within the same asymptotic running time bound.

For the general SMT problem (where the topology is unknown), well-known techniques from the Euclidean and rectilinear problems could be adapted to the fixed orientation problem. However, the adaptation of the Geosteiner approach would not have been particularly successful without the characterization provided by canonical forms.

One of the future challenges is to fill the gap between the polynomial-time approximation scheme of Arora [5], and the plethora of practical heuristics for the problem. It would be useful to have practical heuristics with parameterized approximation bounds.

4 Applications and generalizations

In the first decades of very large scale integrated (VLSI) chip production Manhattan routing was the standard. Two layers were used for (signal) wiring, one for horizontal and one for vertical wires; vias were used to connect wires running in perpendicular directions. (The de-facto rule that wires on a single layer all run in the same direction is sometimes denoted the *preferred* direction constraint.)

During the 1990s advances in manufacturing technologies made it possible to produce chips with more than two interconnect layers. The increased number of layers opened up the possibility of using alternative directions as to improve the quality of the routing with respect to congestion, delay and power usage. As a general rule, decreasing total wire length improves all these quality measures, and may even lead to a reduced chip size.

In this section we first discuss some early contributions related to non-Manhattan routing. The application to printed circuit boards and channel routing is described first. Then we move on to discuss advantages and disadvantages of pervasive use of non-Manhattan wiring, that is, for general full-chip routing. We conclude the section by describing a number of selected generalizations of the fixed orientation Steiner tree problem that are motivated by applications in chip design.

4.1 Printed circuit boards and channel routing

The first application of non-Manhattan routing in printed circuit design appears to be due to Heiss [92]. In 1968, he gave an extension of the classical Lee algorithm [120] that enabled *diagonal* routing (Figure 17). Heiss also gave a generalization to more than two routing layers (where two layers correspond to the two surfaces of a double sided printed circuit board). More recently, Stan et al. [191] presented two other generalizations of Lee's algorithm.

Another alternative to Manhattan routing was given by Chaudhuri [38] in 1979. Chaudhuri introduced routing with three uniform orientations (corresponding to $\lambda = 3$) for both printed circuit boards and channel routing. The new metric was denoted the "Steiner metric" to distinguish it from the usual rectilinear (or Manhattan) metric. For printed circuit boards, Chaudhuri described a general routing scheme for two layers, and he presented a method to deal with the problem of routing three orientations on only two layers.

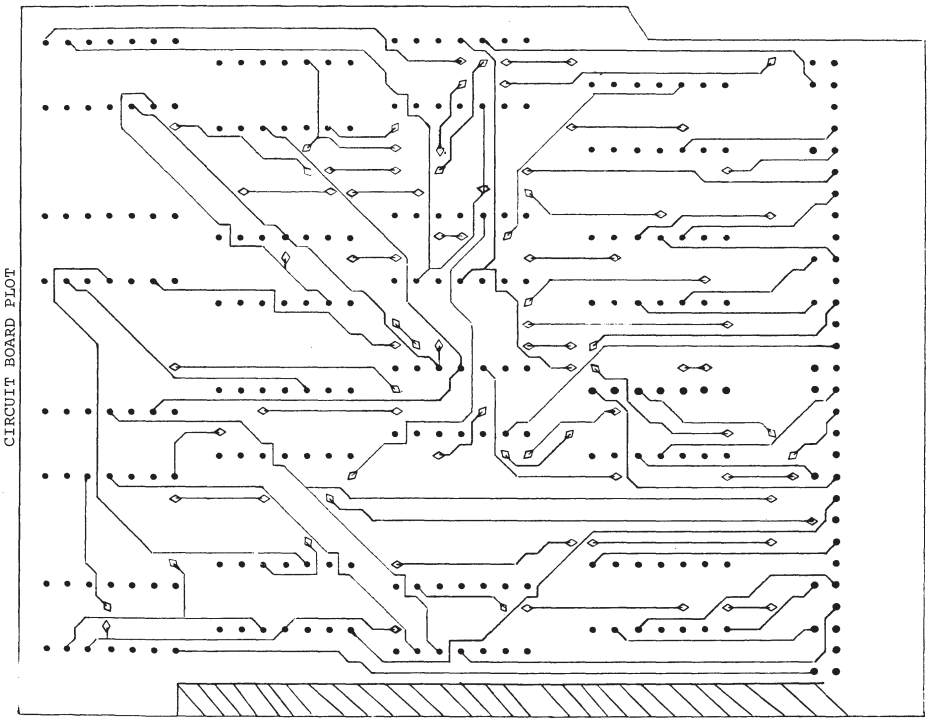


Figure 17: Printed circuit board with diagonal routing (from Heiss [92]).

The other problem that Chaudhuri considered was the *channel routing problem*. Channel routing was one of the basic problems in chip design up to the late 1990s. In the technology of the time, cells were placed on rows on the chip surface, and routing was performed in the area between these rows (so-called channels). A channel consists of two horizontal shores, where the terminals to be interconnected are located. Usually, each net consists of terminals from both shores; hence the interconnection for each net has to cross the channel.

In the traditional *Manhattan routing model*, routing is performed on a rectilinear grid with horizontal *tracks* and vertical *columns* (Figure 18). The number of tracks is called the *width* of the channel. The main objective of the channel routing problem is to minimize the number of tracks needed for the routing as this minimizes the area used by the channel. This is in general an NP-hard problem. Since the objective is not to minimize the length of the nets (individually or jointly), channel routing may result in long connections for some of the nets. For a detailed

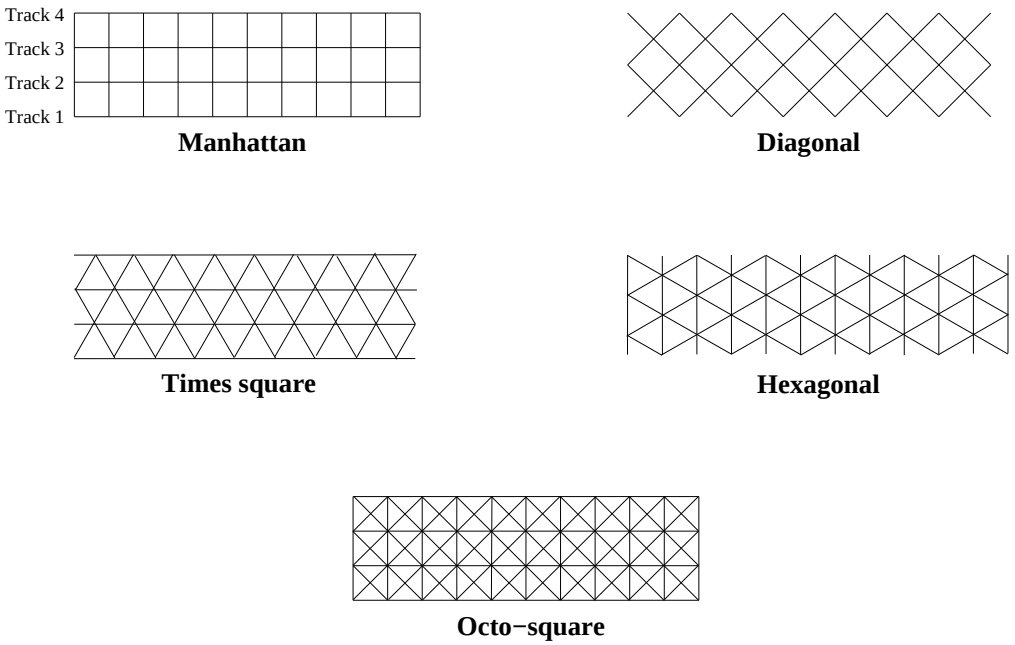


Figure 18: Channel routing models: Manhattan, diagonal, times square, hexagonal and octo-square.

introduction to the theory and algorithms for the Manhattan model, see [9, 126]. In the following we discuss some of the alternatives to the Manhattan model that have been considered in the literature.

In the *diagonal routing model*, the rectilinear grid is simply rotated $45°$ — hence there are still only two orientations [140, 141, 143] (Figure 18). One of the immediate advantages of this model is that short two-terminal interconnections only require one layer change as compared to the Manhattan model where two changes always are needed — except when the two terminals can be connected by a direct vertical connection. Thus fewer vias are in general needed under this model.

Returning to the problem studied by Chaudhuri [38], where three uniform orientations are employed, one may distinguish between two cases. In the first case, the so-called *times square model*, one of the three orientations is the (usual) horizontal track, and the other two orientations are denoted right and left tracks, respectively [20, 142, 190, 195, 196]. In the second case, the *hexagonal model*, one of the three orientations is vertical [21, 166] (Figure 18). One of the advantages of the latter is that terminals can be spaced at the same interval that separates the

wires in the grid; this is not possible under the times square model. Both models can be shown to have superior properties when compared to the Manhattan or diagonal models.

Finally, the *octo-square model* merges the Manhattan and diagonal models. This model thus has four uniform orientations (corresponding to $\lambda = 4$, see Figure 18). This model is discussed in a series of papers [9, 37, 39, 125, 153, 177, 205, 212, 226]. Although this model clearly has the advantage of having more available orientations, it also has the disadvantage that the rectilinear and diagonal wires cannot possibly have the same separation. A slightly different model was discussed by Chiang and Sarrafzadeh [45], who introduced 45° wires locally to avoid so-called knock-knees in the wiring.

One particular subproblem which appears in the channel routing problem — as well as in a more general setting — is the so-called *wiring* problem: Assuming a layout (of some nets) in the plane, assign each edge of the layout to a unique layer such that the connectivity of the nets is preserved, and edges of distinct nets do not overlap on the same layer. Lipski and Preparata [138] presented the first systematic treatment of this problem, and gave a simple characterization of two-layer wirability of arbitrary layouts. Tollis [200, 201] extended the theory to all *uniform* grids; these are grids where the degree of each grid point is even, and where the edges are uniformly distributed around each grid point (there exist exactly four such uniform grids — corresponding to two, three, four and six uniform orientations).

It is unclear to what extent the proposed models and algorithms have found their way into the design of real chips. Channel routing essentially became obsolete during the 1990s as a result of the new sea-of-cells technology, where cells could be placed (more or less) freely on the chip surface; also, over-the-cell routing became possible. In the next subsection we discuss general non-Manhattan routing which is relevant for current day technology.

4.2 General routing in chip design

The application of multiple orientations to the general routing problem in chip design was already anticipated by Widmayer et al. [214, 215] in 1985. In the early 1990s, Burman et al. [33] and Sarrafzadeh and Wong [180] gave the first practical applications of λ-geometry to general routing in chip design. During the

following decade a series of heuristics — mostly with inspiration from rectilinear and Euclidean counterparts — were proposed for solving the Steiner tree problem in uniform and fixed orientation metrics (see Section 3.3).

In 2000 Koh and Madden [116] presented the first in-depth study of the feasibility of large-scale non-Manhattan routing architectures. Using simulation on realistic benchmarks they showed that average wire length reductions between 1% and 11% could be obtained for hexagonal routing ($\lambda = 3$) for a complete chip. Similarly, reductions between 6% and 17% could be obtained for octilinear routing ($\lambda = 4$). It should be noted that these improvements were obtained from the same placement of cells on the chip.

Choi et al. [47] presented a similar analysis for octilinear routing that confirmed the reductions in wire length; however, these reductions were obtained at the cost of an increase in the number of non-routed nets and an increase in the number of vias.

X architecture

The commercial interest in non-Manhattan routing can be witnessed with the formation of the X-Initiative in 2001 [222], a consortium of software and chip companies that supported the development of the so-called *X architecture* [197, 198]. The X architecture essentially adds diagonal wires to traditional Manhattan architecture (corresponding to $\lambda = 4$). However, in order to make this work in practice, a number of difficulties had to be adressed [102].

One of the major problems with the X architecture is that gridded routing does not work in practice; rectilinear and diagonal wires do not have the same separation (see Figure 18, octo-square model). This either results in problems with signal integrity or delay (if diagonal wires are too close or too thin), or results in suboptimal use of routing area (if only every second diagonal wire is used or the separation between rectilinear wires is increased). One solution to this problem — but an algorithmically challenging one — is to drop the preferred direction constraint and to allow all directions on all layers; this is denoted *liquid routing*. This allows for directional changes on a single layer, and can dramatically reduce the number of vias.

Ho et al. [96] suggested a multilevel approach for the X architecture. A multilevel algorithm consists of two main steps: coarsening followed by uncoarsening. The

coarsening step is similar to the use of a global routing algorithm, but is iteratively employed. The algorithm of Ho et al. on average reduced wire length by 18.7% for a set of benchmark instances when compared to Manhattan routing.

Y architecture

As a reaction to the shortcomings of the X architecture, Chen et al. [40, 41, 42, 43] in 2003 took one step back and investigated the use of hexagonal routing (corresponding to $\lambda = 3$); they coined this the *Y architecture*. The advantage is that gridded routing is in fact possible for this architecture since all parallel wires have the same separation (see Figure 18, times square or hexagonal model). Therefore, from an algorithmic point of view, this architecture has a major advantage over the X architecture. Based on simulations under realistic scenarios, Chen et al. [42] estimated that the Y architecture improves wire length in the range $5-8\%$ over Manhattan architecture, while the X architecture obtains improvements in the range $9-11\%$. If the effect of decrease in routing area can be fully utilized to make the chip smaller, wire length improvements of approximately 23% and 29% are possible for Y and X architecture, respectively.

Although it is clear that both the X and Y architecture have the potential to decrease wire length significantly, it is (as of this writing) unclear how many chips have been produced with these architectures. The first commercial chip (from Toshiba[1]) using the X architecture was produced in 2004, and at least one chip (from ATI[2]) followed in 2005.

General architecture

In *general architecture* any number of uniform orientations can be used for routing. Since each routing layer has a preferred direction, the total number of available orientations usually depends on the number of layers. However, since more than 10 routing layers are already common, the number of available layers is not a limiting factor.

The problem of balancing the use of routing resources on the available layers was studied for the Manhattan architecture by Yildiz and Madden [227, 228], and for

[1] www.semiconductor.net/article/CA513393.html
[2] www.edn.com/article/CA608028.html

general architecture by Agnihotri and Madden [1]. The idea is to make the routing cost on each layer (as seen by the routing algorithm) depend on the congestion on the layer. By iteratively adjusting the routing cost on each layer, congestion can be lowered on highly utilized layers.

Paluszewski et al. [158, 159] presented a completely different approach to deal with congestion when many routing layers are available. The idea is to exploit the fact that SMTs usually can be embedded in many different ways in λ-geometry.

The novel idea suggested by Paluszewski et al. [158, 159] is to use a pure geometric approach for routing. In the first phase of the algorithm, SMTs and their *flexibility polygons* are computed for each net on the chip; recall that a flexibility polygon is a geometric representation of all SMTs for a given net (see Section 3.2). Each flexibility polygon is assigned a weight that is equal to the routing area used by the SMT divided by the area of the flexibility polygon (hence a degenerate flexibility polygon with no flexibility has weight 1). Note that the weight represents the average probability that routing resources are needed for a given point in the flexibility polygon.

Based on the weights of the flexibility polygons, a *congestion map* is constructed for the whole chip area (Figure 19). The congestion map gives the estimated routing resources needed for each point of the chip area. The idea of the algorithm is now to move wires away from highly congested areas. This is done by picking a flexibility polygon with least flexibility in a highly congested area and performing the actual routing for this SMT — and in such a way that it avoids congested areas if at all possible. This is repeated until all SMTs are routed, and results in a so-called *initial routing*.

The next step of the algorithm is to remove overlapping wires that still may be left in the initial routing. This is achieved using traditional Maze routing techniques. Finally, liquid routing is applied to reduce the number of vias. This means that wires may be assigned to non-preferred layers. Experiments with the new method show that it is indeed an advantage to use flexibility polygons in the initial routing phase. The initial routing has significantly less overlap than one produced using a standard routing technique, and the final routing also uses less wire length. When using 5 or 6 layers, architectures with $\lambda \geq 4$ reduce total wire length with $7-18\%$ when compared to Manhattan routing. Thus, this prototype implementation is a witness of the potential of applying a general architecture to the routing problem in chip design.

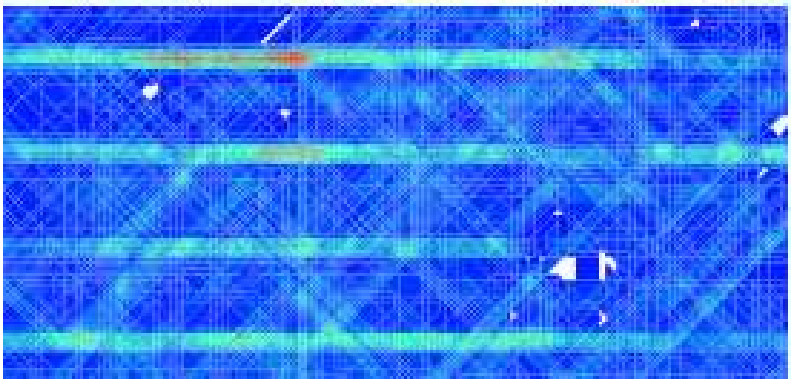


Figure 19: Congestion map (from [158]). Blue and white areas have little congestion, green areas some congestion, while red areas have high congestion.

4.3 Generalizations motivated by chip design

In this section we present a number of generalizations of the Steiner tree problem that are motivated by chip design. The presentation is not comprehensive and does not cover all the literature, but it does cover the most important generalizations. Other generalizations can be found in, e.g. [215].

Wire length estimation

In the placement phase of chip design, wire length can be estimated using a number of different *netlength models*. The purpose of a netlength model is to quantify the quality of a given placement; total wire length is usually a primary objective, but other objectives such as signal delay and power consumption also play an important role.

Each net of the chip (in Manhattan architecture) is a rectilinear tree spanning a set of terminals N. A rectilinear SMT obviously has minimal length, but is NP-hard to compute. Therefore, computational methods that quickly can estimate SMT length — and that can be incorporated directly and efficiently into the objective function of the placement algorithm — are of great interest.

Consider the smallest axis-parallel rectangle that contains N, the so-called *bounding box* $BB(N)$ of N (Figure 20, left). Let $|BB(N)|$ be the perimeter length of $BB(N)$. In the *rectilinear metric* it is easy to see that the half-perimeter length

$\frac{1}{2}|BB(N)|$ is a lower bound on the SMT length for N. Also, the placement problem under this model can be solved by linear programming; in fact, it is a solution to the dual of a transshipment problem [32].

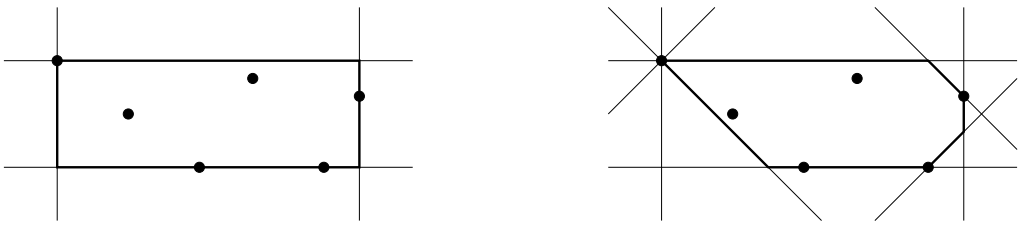


Figure 20: Bounding boxes for $\lambda = 2$ (left) and $\lambda = 4$ (right) for the same set of terminals. Each of the lines (in legal orientations) that define the bounding boxes are shown.

Consider the worst-case relative error $\overline{\rho}(n)$ of the half-perimeter estimate for a set of n terminals under the rectilinear metric. It is easy to see that $\overline{\rho}(2) = \overline{\rho}(3) = 1$, that is, the estimate is exact for $n = 2$ and $n = 3$. Chung and Hwang [51] studied the properties of $\overline{\rho}(n)$ for small n. Using fairly involved arguments they showed that $\overline{\rho}(n) \leq 2$ for $n \leq 10$ (i.e., the SMT length is at most twice the value of the lower bound for $n \leq 10$). More generally, $\overline{\rho}(n)$ grows as $\Theta(\sqrt{n})$ [50, 51]; the best known upper bound is $\frac{1}{2}\lceil\sqrt{n-2}\rceil + \frac{3}{4}$ and is due to Brenner and Vygen [31]. Therefore, the half-perimeter estimate becomes arbitrarily bad in the worst-case as n increases. (For uniformly distributed terminals, the expected error is also $\Theta(\sqrt{n})$ in the limit [84].)

Now consider an arbitrary (weighted) fixed orientation metric with σ legal orientations given by unit circle $\mathcal{C}$. Let $BB_{\mathcal{C}}(N)$ be a smallest convex polygon with sides in σ-geometry that contains N. The polygon is a "constrained" convex hull of N where the sides have restricted orientations; note that $BB_{\mathcal{C}}(N)$ has minimum perimeter among all such polygons. This generalized bounding box can be computed in $O(\sigma n)$ time as follows. For each of the 2σ legal directions, identify a terminal t such that the right half-plane (as seen from t and relative to the given direction) includes all terminals. Clearly, terminal t can be identified in $O(n)$ time. The intersection of these 2σ half-planes defines $BB_{\mathcal{C}}(N)$ (see Figure 20).

As in the rectilinear case, the half-perimeter length $\frac{1}{2}|BB_{\mathcal{C}}(N)|$ is exact for $n = 2$, but not in general for $n > 2$. Furthermore, we have the following:

Theorem 4.1 *The half-perimeter length $\frac{1}{2}|\mathrm{BB}_{\mathcal{C}}(N)|$ is a lower bound on the SMT length for N under any fixed orientation metric.*

Wei, Dong and Hong [210] experimentally validated this claim for $\lambda = 3$, but provided no proof. Here we sketch a proof of the theorem (see also [169] for a similar proof for the Euclidean case). Let T be an SMT for N. Consider an outer walk of T. This outer walk visits the terminals on the boundary of $BB_{\mathcal{C}}(N)$ in order along the boundary. Consider two consecutive boundary terminals u and v. The main observation is that the path between u and v along the boundary is a *shortest path* in σ-geometry. Since the path in T between u and v is at least the shortest distance between u and v, we have $|BB_{\mathcal{C}}(N)| \leq 2|T|_{\mathcal{C}}$. Thus the theorem follows.

It is easy to see that the asymptotic results for the rectilinear metric generalize to arbitrary λ-geometry, since λ-metrics are bounded by constant factors from each other. The same holds for unweighted fixed orientation metrics. For arbitrary *weighted* fixed orientation metrics, the constants in the asymptotic bounds depend on the skewness of the unit disk.

One straightforward way to improve the half-perimeter estimate is to scale it with a function that depends on n. More precisely, a function of the form $C\sqrt{n}$, where C is an appropriate constant, is typically chosen. The constant factor C is found using statistical methods by performing experiments on uniformly distributed and/or real-life problem instances. Wei, Dong and Hong [210] studied the half-perimeter estimate for $\lambda = 3$. They developed an improved wire length estimation method denoted APWL-Y by performing more sophisticated scaling of the half-perimeter estimate. In their method, the scaling not only depends on n, but also on the "aspect ratio" of the bounding box, that is, the amount of skewness of the bounding box. Based on comprehensive experiments, Wei et al. [210] conclude that the error of the improved wire length estimate is less than 5% on average. This error is similar to the error obtained by computing a minimum spanning tree (MST) for the terminals; however, computing the wire length estimate is much faster than computing an MST.

A number of other methods can be used to esimate wire length. For the rectilinear metric, Brenner and Vygen [31] compare the following alternative estimates: MST length [171], star length (solution to the general Fermat problem for N) and clique length (sum of distances over all pairs of points divided by some function of n). Brenner and Vygen [31] argue that the clique model is most appropriate in placement algorithms that only consider two-terminal connections, that is, where a fixed topology must be assumed for each net of the chip.

Delay-driven routing

In the chip routing problem the task is to interconnect the terminals of every net of the chip. For each net N, one of the terminals $r \in N$ is the *source*, while the remaining terminals in N are the *sinks*. The electrical signal should propagate from the source to the sinks via the constructed tree. Thus the Steiner tree is in fact directed — or a so-called *Steiner arborescence*.

One of the important objectives that should be taken into account in the routing phase of a net is the *signal delay* from the source to the sinks. In particular, if the net is part of the *critical signal path* of the chip, then the signal delay of the constructed tree has a direct influence on the clock-rate (or performance) of the chip. Signal delay is related to the length of the paths from the source to the sinks, so minimizing total path length often improves the signal delay properties of the tree. However, the actual signal delay has more complex behaviour, and depends not only on the length of the path itself, but also on the length of the subtrees that are rooted on the path. Furthermore, for a simple two-terminal connection, signal delay increases quadratically with the length of the connection. The popular Elmore delay model [74, 86, 113, 161] serves as a good estimation for computing the signal delay. The slightly simpler distributed RC delay model is easier to use for optimization, but serves best as an upper bound on the delay [65]. An overview of models and techniques for optimizing delay can be found in [63, 113]. The problem of minimizing Elmore delay is still a major algorithmic challenge in chip design [117].

In the remainder of this subsection we consider some simpler — but still useful — models that can incorporate delay into Steiner tree algorithms. The simplest model is to assume that the delay of a wire is linear in its length. Ignoring all other objectives, a shortest path tree would provide minimum delay. However, shortest path trees usually have unacceptable high total length. (Note that a shortest-path tree in the Euclidean metric is a star centered at the source.) Therefore, so-called *shallow-light* trees [7] have attracted considerable interest. These are trees that both have short paths from their source to their sinks (are “shallow”) and that have small total length (are “light”).

The two objectives, path length and total length, can be combined in a number of different ways, e.g., by bounding both objectives, by bounding one objective while minimizing the other, or by minimizing a weighted sum of both objectives. This can be achieved by combining Prim’s algorithm for constructing minimum

spanning trees with Dijkstra's algorithm for constructing shortest path trees [3, 64, 114, 151]. Shallow-light Steiner trees can also be obtained by adding appropriate shortest paths to a heuristic SMT [64]. Most of the algorithms for constructing shallow-light trees can easily be extended to any fixed orientation metric as they are in their core graph based.

For most nets on the chip, an SMT has sufficiently good delay properties. Furthermore, SMTs in fixed orientation metrics are usually not unique (see Section 3.2), and the SMTs for a given terminal set may have varying delay properties (Figure 21).

Figure 21: Two rectilinear SMTs for the same set of terminals. The SMT on the right has better signal delay properties than the SMT on the left.

The construction of (short) rectilinear Steiner trees that are good with respect to some other objective than length was considered by Boese et al. [11, 12, 13] and by Bozorgzadeh et al. [18, 19]. Boese et al. introduced a *Global Slack Removal* algorithm that attempts to improve the delay properties of a Steiner tree without increasing its length. Peyer et al. [163] took this idea one step further and considered the problem of constructing rectilinear SMTs with some *secondary* delay-related objective. More specifically, they focused on the problem of constructing a rectilinear SMT with the weighted sum of path lengths as the secondary objective (i.e., path lengths from a given source to a set of sinks). An optimal solution to this problem exists in the Hanan grid for the terminal set [232], but Peyer et al. [163] proved the following stronger result:

Theorem 4.2 *The Steiner points of an optimal solution to the rectilinear Steiner tree problem with weighted sum of path lengths secondary objective* must *overlap with vertices of the Hanan grid for the terminal set.*

Adding the secondary objective to the problem thus forces the Steiner points to belong to the Hanan grid. Contrary to the rectilinear problem, no structural results are known for the same problem under other fixed orientation metrics; however, it is easy to see that every FST of an optimal solution can be embedded in such

a way that the FST has at most one bent edge (see Theorem 2.6). We conjecture that there exist strong characterizations based on the concept of canonical forms (Section 3.1).

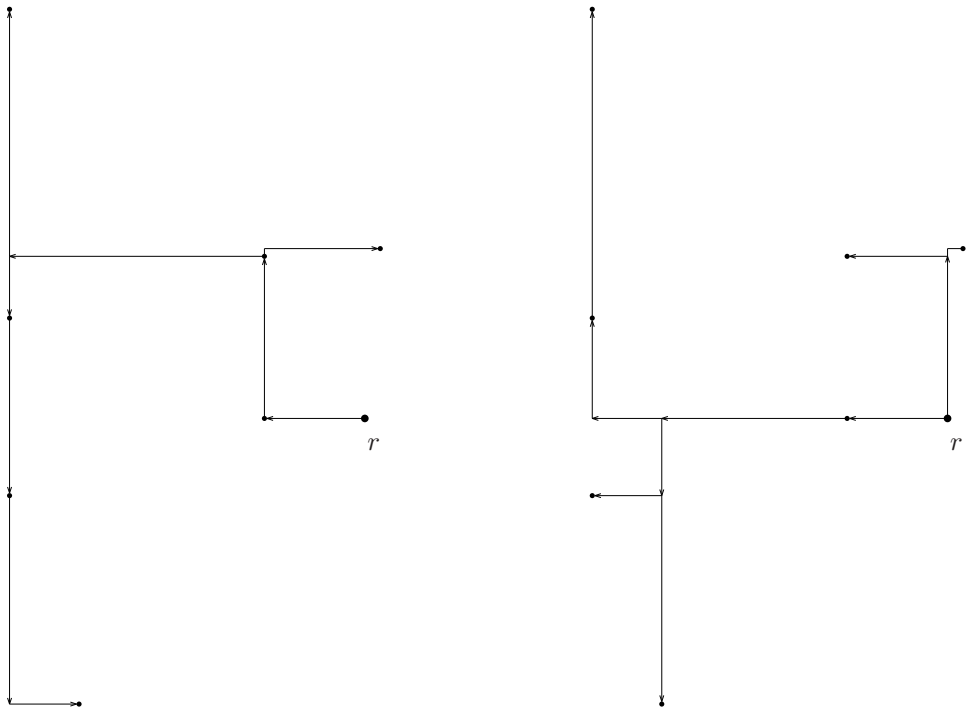
For the rectilinear problem, Peyer et al. [163] presented both exact and heuristic algorithms. The heuristic algorithm, denoted *Extended Global Slack Removal (XGSR)*, is also capable of minimizing Elmore delay as a secondary objective. Experiments with real-life chip instances with 4 to 40 terminals were presented. (Note that rectilinear SMTs with 2 or 3 terminals are always optimal wrt. weighted sum of path lengths.) On average, XGSR constructed secondary objective optimal trees for 98.4% of the problem instances; only 52.0% of the problem instances were optimal before applying XGSR. Figure 22 shows two real-life rectilinear SMTs — one of which is secondary objective optimal.

Group interconnections

During the routing phase of chip design, it is usually assumed that each pin of the net is a single point. Thus computing a minimum length interconnection is the same as computing a Steiner minimum tree for the pins of the net. However, on a real chip a pin typically consists of several rectangles (or line segments), and any point on this set of rectangles suffices as a connection point. This fact motivates the study of so-called *group* Steiner trees, where each "terminal" consists of a set of rectangles. The roots of this problem go all the way back to Melzak [146], who discussed the Euclidean group problem (where the groups are convex sets in the plane).

As shown by Zachariasen and Rohe [235], the rectilinear group problem with rectangles reduces to the rectilinear group problem with points, as the problem can be solved in the Hanan grid of the corner points of the given rectangles. So from here on we consider the following definition of the rectilinear group Steiner tree problem: Given a set of groups $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$, where each group N_i, $i = 1, \ldots, k$, is a finite set of points in the plane, construct a shortest rectilinear tree which spans at least one point from each group. Such a tree is called a *rectilinear group Steiner minimum tree (RGSMT)*. A real-life RGSMT is shown on Figure 23.

This problem is NP-hard even for very restricted cases, e.g., when all the terminals are required to lie on two parallel lines [103] or when Steiner points are not allowed [104]. In contrast to the ordinary Steiner tree problem, no polynomial-



SMT (503) XGSR (359)

Figure 22: Real-life chip net example with 8 terminals. Both trees are rectilinear SMTs. The weighted sum of path lengths is given for each tree. The tree obtained by XGSR is secondary objective optimal.

time approximation scheme or even constant-approximation algorithm is known for the problem. For the more general group Steiner tree problem in *graphs* it is known that the problem cannot be approximated better than a factor $\ln k$ in polynomial time, assuming P $\neq$ NP [75, 104]. However, this does not exclude the existence of better approximation algorithms for geometric variants of the problem. Scultze [183] gave a constant-factor algorithm for the Euclidean problem where the groups are contained in disjoint regions with bounded "fatness" (disks or squares have for example bounded fatness).

Practical heuristics for the rectilinear group Steiner tree problem were first considered by Reich and Widmayer [172]. These algorithms were experimentally eval-

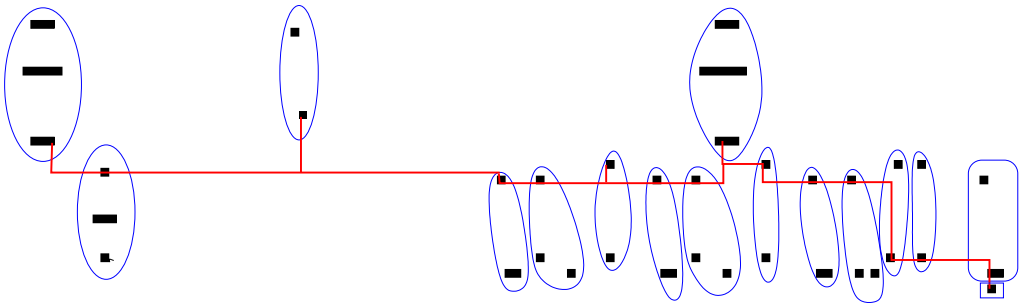


Figure 23: Rectilinear group Steiner minimum tree (real-life chip instance from [235]). The line segments within each group (marked by a "bubble") are electrically equivalent, and the tree should span at least one point from each group.

uated in [104] and appeared to compute good solutions. Other practical heuristics showing good performance have been proposed [8, 93].

As mentioned above, the rectilinear group Steiner tree problem can be reduced to the group Steiner tree problem in the Hanan grid graph for the given points. Furthermore, it is easy to transform the group problem to the ordinary Steiner tree problem in a graph by introducing so-called super-terminals to the graph problem. Therefore, the problem can be solved using any exact algorithm for the Steiner tree problem in graphs.

Zachariasen and Rohe [235] gave a first (tailored) exact algorithm for solving the rectilinear group Steiner tree problem. They presented techniques to reduce the given set of points, that is, to remove points in the groups N_i from consideration by showing that an RGSMT exists that does not use these points. Also, a generalized version of Zachariasen's [231] full Steiner tree generation algorithm was used to reduce the Hanan grid graph — hence speeding up standard branch-and-cut algorithms for solving the corresponding graph problem. Computational experiments on real-life and random problem instances with up to 100 groups were performed. The techniques employed resulted in a speed-up approaching an order of magnitude, and increased the range of practically solvable real-life problem instances from around 40 groups to beyond 70 groups.

Except from the above mentioned work on the rectilinear problem, no literature appears to exist on the group problem for fixed or uniform orientation metrics. One particular difficulty is that no computationally efficient reduction to a graph problem exists, as the ordinary problem does not reduce to a polynomially-sized graph problem (see Section 2.3). From an approximation algorithm point of view,

it is likely that all fixed orientation metrics have the same asymptotical upper and lower bounds on their approximation ratio. However, as pointed out above, no non-trivial/specialized upper or lower bounds are currently known.

Obstacle-avoiding interconnections

When solving the routing problem in chip design, certain regions of the chip surface may be forbidden — or may have certain restrictions. Such regions are usually denoted *obstacles*. Obstacles typically consist of pre-placed macros or other circuits (Figure 24). In older technologies, where the number of available layers was limited, routing across pre-placed circuits was impossible. These circuits formed *hard* obstacles. In newer technologies, where the number of layers is higher, it is possible to route wires across pre-placed circuits. However, the length of wires across such *soft* obstacles must usually be bounded, since there is no room for placing buffers/repeaters in areas with obstacles [148].

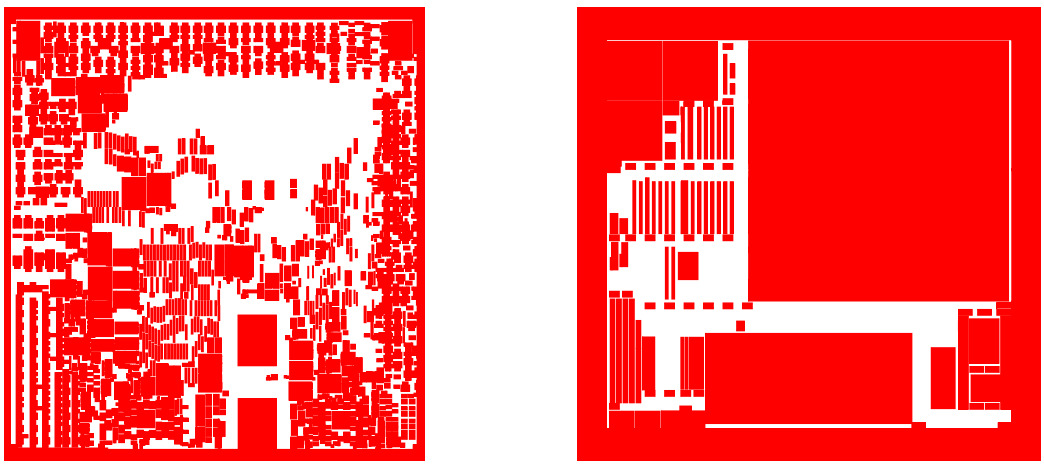


Figure 24: Typical distribution of obstacles on two real-life chips. (Pictures reproduced by courtesy of Research Institute for Discrete Mathematics, University of Bonn.)

An obstacle-avoiding Steiner minimum tree (OSMT) is a tree that interconnects a given set of terminals using minimum total length while avoiding a set of given hard obstacles (Figure 25). The problem with hard, polygonally-bounded obstacles has by far received the most attention in the literature, and in the following we focus on this particular problem. (Studies of other variants can be found in [121, 124, 148].)

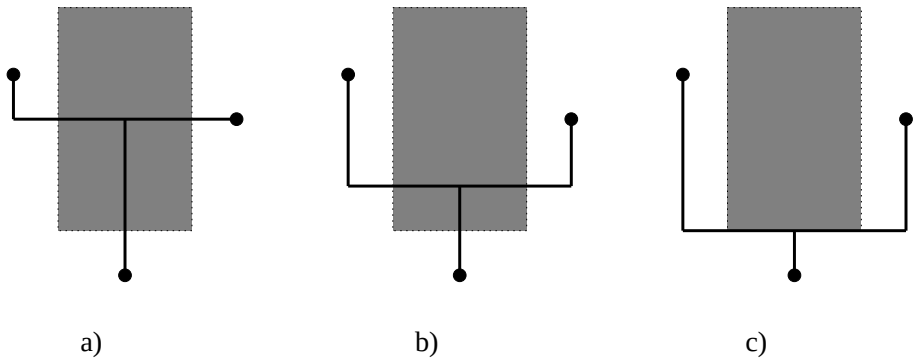


Figure 25: Soft and hard obstacles. a) A rectilinear SMT that ignores an obstacle. b) A rectilinear SMT that respects a soft obstacle by bounding (the length of) the part of the tree that overlaps with the obstacle; note that the tree is not an SMT for terminals. c) An hard obstacle and an obstacle-avoiding rectilinear SMT.

The construction of obstacle-avoiding *shortest paths* forms a building block in the construction of obstacle-avoiding Steiner trees. A shortest obstacle-avoiding path can be found in the so-called visibility graph [6, 211], which has the set of terminals and obstacle corners as vertices, and an edge between two vertices if it does not properly intersect one of the given obstacles. Wu et al. [221] defined the so-called track-graph for the rectilinear problem, and showed that obstacle-avoiding rectilinear shortest paths can be computed efficiently using this graph. Widmayer [213] gave an $O(n \log n)$ time construction of a rectilinear shortest path preserving graph of size $O(n \log n)$, where n is the number of terminals and obstacle corners.

Ganley and Cohoon [78] observed that the obstacle-avoiding rectilinear Steiner tree problem can be solved in a subset of the Hanan grid (see Section 1.2) of the terminals and obstacle corners — thus providing an efficient reduction to the Steiner tree problem in *planar graphs*. Müller-Hannemann and Schulze [149] also gave an approximative reduction from the obstacle-avoiding *octilinear* Steiner tree problem to the Steiner tree problem planar graphs. For the Euclidean problem, Provan [169] gave an approximative reduction to an almost-planar graph problem and derived a fully polynomial-time approximation scheme (FPTAS) for the special case where the terminals lie on a constant number of boundary polygons and interior points. Provan also defined the so-called path-convex hull — as a generalization of the convex hull of the terminals in the obstacle-free case — and showed that there exists an OSMT in this region. Winter [217] extended the notion of

visibility graphs to accomodate the construction of Euclidean OSMTs.

In addition to these graph problem reductions, a number of designated heuristics and exact algorithms have been considered for the obstacle-avoiding problem. Winter and Smith [219] gave a polynomial-time algorithm for the construction of a Euclidean OSMT for 3 terminals and one convex obstacle. Zachariasen and Winter [237] experimented with an exact algorithm based on the Geosteiner approach (see Section 3.4), and were able to solve Euclidean problem instances with up to 150 terminals.

For the rectilinear problem, Ganley and Cohoon [78] made experiments with a greedy Steiner point insertion heuristic. Feng et al. [76] and Jin et al. [109] devised $O(n \log n)$ time heuristics for the obstacle-avoiding problem in λ-geometry.

The polynomial-time approximation scheme (PTAS) of Arora [5] for the ordinary Steiner tree problem does not work when hard obstacles are present. It was for a while an open problem whether a PTAS existed for the obstacle-avoiding problem. Recently, Borradaile et al. [16, 17] showed that the Steiner tree problem in planar graphs admits a PTAS. Müller-Hanneman and Tazari [150] used this result to design a PTAS for the obstacle-avoiding problem under any uniform orientation metric. (It appears that this approach also works for any fixed orientation metric.) The result is obtained by reducing the geometric problem to an approximatively equivalent problem in a planar graph and applying the PTAS of Borradaile et al.

Rotational Steiner tree problem

Given some fixed orientation metric, suppose that we are permitted to rotate the coordinate system (or the unit circle), that is, to rotate all legal orientations simultaneously. Let the rotation angle be $\alpha \in [0, \pi[$. What is the value of α minimizing the length of the SMT?

Clearly this *rotational Steiner tree problem* is a harder problem than the ordinary Steiner tree problem, since the ordinary problem just corresponds to solving the problem for $\alpha = 0$. In this section we give some structural properties of *rotationally optimal SMTs* which are trees that have minimum length over all possible $\alpha \in [0, \pi[$. (In fact, it is easy to see that we need only consider the interval $[0, \pi/\lambda[$ in λ-geometry due to the symmetry of the unit circle.) The main result for general σ-geometry is the following:

Theorem 4.3 *In σ-geometry, a rotationally optimal SMT has at least one FST that has no bent edges and that uses at most* three *orientations.*

The proof works by contradiction. If all FSTs contain bent edges, then the length of the tree as a function of α turns out to be *strictly concave*. This property was proved by Nielsen et al. [156] for the rectilinear problem; the fact that rectilinear FSTs can be assumed to have Hwang-topology (see Section 1.3) enabled a fairly straightforward proof. Similarly, for the problem in λ-geometry where λ is a multiple of 3, the property that the Steiner points of an FST can be assumed to coincide with the Steiner points for the corresponding Euclidean FST [155] made it possible to give a fairly simple proof for this special case [23].

The proof of Theorem 4.3 for general λ requires a more careful study of how the length of an FST changes under rotation of the legal orientations. The proof given by Nielsen et al. [23] is in fact so general that it also covers the fixed orientation problem. It should be noted that the property that the length is a strictly concave function of α only helps to prove that there are no bent edges in at least one FST. In order to prove that there are at most three orientations, the results of Section 2.2 on zero-shifts and direction sets are employed. The three orientations must all have either primary or secondary orientations; for λ being a multiple of 3 this means that there must be at least one FST that has exactly the same embedding as a Euclidean FST with the same topology (i.e., with edges meeting at 120° angles).

Experimental results presented in [23] indicate that it is indeed possible to achieve non-trivial length reductions when rotating the legal orientations. On the other hand, these reductions (as could be expected) become negligible when the number of terminals increases or when the number of legal orientations increases. Nielsen et al. [23] conjectured that the structural results on rotational optimal SMTs may turn out to give further insight into the problem of determining the Steiner ratio in λ-geometry for all λ (see Section 2.2).

Other generalizations

In this last subsection we briefly mention some other generalizations of the Steiner tree problem that have been studied. Melzak [145] and Underwood [203] considered the problem of minimizing length *plus* a constant times the number of Steiner points; by increasing the constant, a tree with fewer Steiner points is obtained. Further generalizations were given by Melzak [145, 146], who studied a

general geometric problem where a weighted sum of the length, sum of degrees of terminals, sum of degrees of Steiner points and number of Steiner points is minimized.

Frommer et al. [77] considered an interconnection problem where the cost of edges depend on their location in the plane (i.e., both on their orientation and on their specific location).

Finally, Sarrafzadeh et al. [178] studied rectilinear *floating* Steiner trees. In this problem multiple Steiner trees share a terminal that is movable. This problem is relevant in the placement phase of chip design. The problem is to find the position of the terminal that minimizes overall tree length.

4.4 Conclusions

One of the major challenges of applying non-Manhattan architectures to routing in chip design is the necessity of *liquid* routing — namely that multiple orientations must be allowed on a single layer in order to reduce the number of vias. Without liquid routing, the advantages of multiple orientations are quickly eaten up by layer change costs.

One particular problem for the X architecture, which is not present in the Manhattan or the Y architecture, is that gridded routing does not work — the orientations do not have the same separation. Thus traditional grid-based routing algorithms must be abandoned.

Despite these challenges, there is no doubt that future integrated circuits are going to use more than two orientations for routing — in one way or the other. Hence the theory and the algorithms developed will most likely play an important role in the future.

Final remarks

The Steiner tree problem in Minkowski (and other) spaces was discussed in the book by Hwang, Richards and Winter from 1992 [101, page 287]. One interesting remark is the following:

"What has been missing in the literature is the invention of efficient algorithms to construct a full SMT for a given topology, like what the Melzak FST algorithm does for the Euclidean plane. ... Therefore, finding these efficient algorithms for small numbers of terminals becomes a priority task to attack the Steiner tree problem for general metric spaces."

In Brazil's survey from 2001 on the uniform orientation problem [22], the first of three open problems is the following:

"Given $\lambda > 3$ and a terminal set N, does there exist a polynomial-time algorithm for finding a Steiner minimum λ-tree for any given Steiner topology on N?"

The contributions of this dissertation answer Brazil's question in the affirmative, and even generalize the result to arbitrary (weighted) fixed orientation problems. The dissertation is also a significant contribution to the general problem mentioned by Hwang, Richards and Winter.

The second open problem mentioned by Brazil is related to the generalized Hanan grid (see our Open Problem 2). Our dissertation gives no direct answers to this question, but it is likely that the theory developed will play a crucial role in solving this open problem. In contrast to Brazil, we conjecture that Theorem 2.8 is tight.

The third and final open problem mentioned by Brazil is on the Steiner ratio for general uniform orientation metrics (see our Open Problem 1). Time will show whether this dissertation has provided significant results for the solution of this problem.

Solving open problems is just one part of the story. Perhaps the most significant contribution of our dissertation is its capability to link theory with algorithms, and to link algorithms with applications. The scope of the dissertation — namely fixed orientation interconnection problems — is not particularly broad, but the results cover the full spectrum from theory to applications.

Open problems

Open Problem 1 Determine the exact Steiner ratio for all λ-metrics (see Section 2.2).

Open Problem 2 Prove that the bound given by Theorem 2.8 is tight, that is, it is sufficient *and necessary* to consider the $n-2$ generalized Hanan grid for a general fixed orientation metric.

Open Problem 3 Prove (or disprove) that the distance between two points under a general (weighted) fixed orientation metric can be computed in constant time. (A logarithmic running time can easily be achieved by performing a binary search on the sorted set of given legal orientations.)

Open Problem 4 Give a $O(n \log n)$ time algorithm to construct a *separable* MST for n terminals under any fixed orientation metric (see Section 3.3).

Open Problem 5 Give a $O(n)$ time algorithm to construct an optimal embedding for a separable MST for n terminals under any fixed orientation metric (see Section 3.3).

Open Problem 6 Prove (or disprove) that the SMT for a set of 10 or more terminals on the boundary of a hexagon under the hexagonal metric ($\lambda = 3$) is a minimum spanning tree for the terminals (see Section 3.3).

References

[1] A. R. Agnihotri and P. H. Madden. Congestion Reduction in Traditional and New Routing Architectures. In *Proceedings of the 13th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 211–214, 2003.

[2] M. Alfaro, M. Conger, K. Hodges, A. Levy, R. Kochar, L. Kuklinski, Z. Mahmood, and K. von Haam. The Structure of Singularities in Φ-Minimizing Networks in $\mathbf{R}^2$. *Pacific Journal of Mathematics*, 149:201–210, 1991.

[3] C. J. Alpert, T. C. Hu, J. H. Huang, and A. B. Kahng. A Direct Combination of the Prim and Dijkstra Constructions for Improved Performance-Driven Global Routing. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 1868–1872, 1993.

[4] S. Arora. Polynomial Time Approximation Schemes for Euclidean TSP and other Geometric Problems. In *Proc. 37th Ann. Symp. on Foundations of Computer Science*, pages 2–13, 1996.

[5] S. Arora. Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems. *J. ACM*, 45(5):753–782, 1998.

[6] T. Asano, T. Asano, L. Guibas, J. Herchberger, and H. Imai. Visibility of Disjoint Polygons. *Algorithmica*, 1:49–63, 1986.

[7] B. Awerbuch, A. Baratz, and D. Peleg. Cost-sensitive Analysis of Communication Protocols. In *Proceedings of the Ninth Annual ACM Symposium on Principles of Distributed Computing (PODC '90)*, pages 177–187, New York, NY, USA, 1990.

[8] C. D. Bateman, C. H. Helvig, G. Robins, and A. Zelikovsky. Provably-Good Routing Tree Construction with Multi-Port Terminals. In *ACM/SIGDA International Symposium on Physical Design*, 1997.

[9] B. Berger, M. L. Brady, D. J. Brown, and T. Leighton. Nearly Optimal Algorithms and Bounds for Multilayer Channel Routing. *J. ACM*, 42:500–542, 1995.

[10] M. W. Bern and R. L. Graham. The Shortest-Network Problem. *Scientific American*, pages 66–71, January 1989.

[11] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins. Rectilinear Steiner Trees with Minimum Elmore Delay. In *ACM Design Automation Conference (DAC'94)*, pages 381–386. ACM, 1994.

[12] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins. Near-optimal Critical Sink Routing Tree Constructions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(12):1417–1436, 1995.

[13] K. D. Boese, A. B. Kahng, and G. Robins. High-Performance Routing Trees With Identified Critical Sinks. In *ACM Design Automation Conference (DAC'93)*, pages 182–187. ACM, June 1993.

[14] M. Borah, R. M. Owens, and M. J. Irwin. An Edge-based Heuristic for Steiner Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13:1563–1568, 1994.

[15] M. Borah, R. M. Owens, and M. J. Irwin. A Fast and Simple Steiner Routing Heuristic. *Discrete Applied Mathematics*, 90:51–67, 1999.

[16] G. Borradaile, C. Kenyon-Mathieu, and P. Klein. A Polynomial-time Approximation Scheme for Steiner tree in Planar Graphs. In *SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1285–1294, 2007.

[17] G. Borradaile, P. Klein, and C. Kenyon-Mathieu. Steiner Tree in Planar Graphs: An $O(n \log n)$ Approximation Scheme with Singly-Exponential Dependence on Epsilon. In *WADS 07: Proceedings of the 10th Workshop on Algorithms and Data Structures. LNCS 4619*, pages 275–286, 2007.

[18] E. Bozorgzadeh, R. Kastner, and M. Sarrafzadeh. Creating and Exploiting Flexibility in Steiner Trees. In *ACM Design Automation Conference (DAC'01)*, pages 195–198. ACM, June 2001.

[19] E. Bozorgzadeh, R. Kastner, and M. Sarrafzadeh. Creating and Exploiting Flexibility in Steiner Trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22:605–615, 2003.

[20] M. L. Brady, D. J. Brown, and K. Powers. Channel Routing on a 60° Grid. In *Proc. Conf. Information Science and Systems*, pages 926–931, 1990.

[21] M. L. Brady, D. J. Brown, and K. Powers. Hexagonal Models for Channel Routing. *Algorithmica*, 19:263–290, 1997.

[22] M. Brazil. Steiner Minimum Trees in Uniform Orientation Metrics. In D.-Z. Du and X. Cheng, editor, *Steiner Trees in Industries*, pages 1–27. Kluwer Academic Publishers, 2001.

[23] M. Brazil, B. K. Nielsen, P. Winter, and M. Zachariasen. Rotationally Optimal Spanning and Steiner Trees in Uniform Orientation Metrics. *Computational Geometry: Theory and Applications*, 29:251–263, 2004.

[24] M. Brazil, D. A. Thomas, and J. F. Weng. Minimum Networks in Uniform Orientation Metrics. *SIAM Journal on Computing*, 30:1579–1593, 2000.

[25] M. Brazil, D. A. Thomas, and J. F. Weng. Forbidden Subpaths for Steiner Minimum Networks in Uniform Orientation Metrics. *Networks*, 39:186–202, 2002.

[26] M. Brazil, D. A. Thomas, and J. F. Weng. Locally Minimal Uniformly Oriented Shortest Networks. *Discrete Applied Mathematics*, 154:2545–2564, 2006.

[27] M. Brazil, D. A. Thomas, J. F. Weng, and M. Zachariasen. Canonical Forms and Algorithms for Steiner Trees in Uniform Orientation Metrics. *Algorithmica*, 44:281–300, 2006.

[28] M. Brazil, P. Winter, and M. Zachariasen. Flexibility of Steiner Trees in Uniform Orientation Metrics. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC), Lecture Notes in Computer Science 3341*, pages 196–208, 2004.

[29] M. Brazil, P. Winter, and M. Zachariasen. Flexibility of Steiner Trees in Uniform Orientation Metrics. *Networks*, 46:142–153, 2005.

[30] M. Brazil and M. Zachariasen. Steiner Trees for Fixed Orientation Metrics. *Journal of Global Optimization*, 43:141–169, 2009.

[31] U. Brenner and J. Vygen. Worst-Case Ratios of Networks in the Rectilinear Plane. *Networks*, 38:126–139, 2001.

[32] U. Brenner and J. Vygen. Analytical Methods in VLSI Placement. In C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, editors, *Handbook of Algorithms for VLSI Physical Design Automation*, chapter 17, pages 327–346. Taylor and Francis, Boca Raton, 2009.

[33] S. Burman, H. Chen, and N. Sherwani. Improved Global Routing using λ-geometry. In *Proc. 29 Annual Allerton Conference on Communications, Computing and Controls*, 1991.

[34] P. B. Callahan and S. R. Kosaraju. A Decomposition of Multidimensional Point Sets with Applications to k-Nearest-Neighbors and n-Body Potential Fields. *J. ACM*, 42:67–90, 1995.

[35] G. D. Chakerian and M. A. Ghandehari. The Fermat Problem in Minkowski Spaces. *Geometriae Dedicata*, 17:227–238, 1985.

[36] T. H. Chao and Y. C. Hsu. Rectilinear Steiner Tree Construction by Local and Global Refinement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(3):303–309, 1994.

[37] K. Chaudhary and P. Robinson. Channel Routing by Sorting. *IEEE Transactions of Computed-Aided Design*, 10:754–760, 1991.

[38] P. P. Chaudhuri. An Ecological Approach to Wire Routing. In *IEEE International Symposium on Circuits and Systems*, pages 854–857, 1979.

[39] C. Y. R. Chen, C. Y. Hou, and U. Singh. Optimal Algorithms for Bubble Sort Based Non-Manhattan Channel Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13:603–609, 1994.

[40] H. Chen, C. K. Cheng, A. B. Kahng, I. I. Mandoiu, and Q. Wang. Estimation of Wirelength Reduction for λ-geometry vs. Manhattan Placement and Routing. In *Proceedings ACM International Workshop on System-Level Intercnnect Orediction (SLIP)*, pages 71–76, 2003.

[41] H. Chen, C. K. Cheng, A. B. Kahng, I. I. Mandoiu, Q. Wang, and B. Yao. The Y-Architecture for On-chip Interconnect: Analysis and Methodology. In *Proceedings ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 13–19, 2003.

[42] H. Chen, C. K. Cheng, A. B. Kahng, I. I. Mandoiu, Q. Wang, and B. Yao. The Y-Architecture for On-chip Interconnect: Analysis and Methodology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24:588–599, 2005.

[43] H. Chen, B. Yao, F. Zhou, and C. K. Cheng. The Y-Architecture: Yet Another On-chip Interconnect Solution. In *Proceedings Asia-Pacific Design Automation Conference*, pages 840–846, 2003.

[44] C. Chiang and C.-S. Chiang. Octilinear Steiner Tree Construction. In *Proceedings of the 45th Midwest Symposium on Circuits and Systems*, pages 603–606, 2002.

[45] C. Chiang and M. Sarrafzadeh. Wirability of Knock-Knee Layouts with 45-Degree Wires. *IEEE Transactions on Circuits and Systems*, 38:613–624, 1991.

[46] C. Chiang, Q. Su, and C.-S. Chiang. Wirelength Reduction by using Diagonal Wire. In *Proceedings of the 13th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 104–107, 2003.

[47] B. Choi, C. Chiang an J. Kawa, and M. Sarrafzadeh. Routing Resources Consumption on M-arch and X-arch. In *Proceedings of the 2004 International Symposium on Circuits and Systems (ISCAS'04)*, volume 5, pages V–73–V–76, 2004.

[48] C. Chu. FLUTE: Fast Lookup Table Based Wirelength Estimation Technique. In *Proceedings ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 696–701, 2004.

[49] C. Chu and Y.-C. Wong. Fast and Accurate Rectilinear Steiner Minimal Tree Algorithm for VLSI Design. In *Proceedings International Symposium on Physical design (ISPD)*, pages 28–35, 2005.

[50] F. R. K. Chung and R. L. Graham. On Steiner Trees for Bounded Point Sets. *Geometriae Dedicata*, 11:353–361, 1981.

[51] F. R. K. Chung and F. K. Hwang. The Largest Minimal Rectilinear Steiner Trees for a Set of n Points Enclosed in a Rectangle with a Given Perimeter. *Networks*, 9:19–34, 1979.

[52] D. Cieslik. The Vertex Degrees of Steiner Minimal Trees in Minkowski Planes. In *Topics in Combinatorics and Graph Theory*, pages 201–206. Physica-Verlag, Heidelberg, 1990.

[53] D. Cieslik. The Vertex Degrees of Steiner Minimal Trees in Banach-Minkowski Spaces. *Geombinatorics*, 3:75–82, 1994.

[54] D. Cieslik. *Steiner Minimal Trees*. Kluwer Academic Publishers, Boston, 1998.

[55] D. Cieslik. *The Steiner Ratio*. Kluwer Academic Publishers, Boston, 2001.

[56] D. Cieslik. The Essential of Steiner's Problem in Normed Planes. Technical Report 8, Ernst-Moritz-Arndt-Universität Greifswald, Preprint-Reihe Mathematik, 2004.

[57] D. Cieslik. *Shortest Connectivity — Introduction with Applications in Phylogeny*, volume 17 of *Combinatorial Optimization*. Springer, New York, 2005.

[58] D. Cieslik. The Steiner-Ratio in Banach-Minkowski-Planes - A Survey. In D. Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization, Supplement Volume B*, pages 55–81. Springer, New York, 2005.

[59] E. J. Cockayne. On the Steiner Problem. *Canadian Mathematical Bulletin*, 10:431–450, 1967.

[60] E. J. Cockayne. On the Efficiency of the Algorithm for Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 18(1):150–159, 1970.

[61] E. J. Cockayne and D. E. Hewgill. Exact Computation of Steiner Minimal Trees in the Plane. *Information Processing Letters*, 22:151–156, 1986.

[62] E. J. Cockayne and D. E. Hewgill. Improved Computation of Plane Steiner Minimal Trees. *Algorithmica*, 7(2/3):219–229, 1992.

[63] J. Cong, L. He, C.-K. Koh, and P. H. Madden. Performance Optimization of VLSI Interconnect Layout. *Integration, the VLSI Journal*, 21:1 – 94, 1996.

[64] J. Cong, A. B. Kahng, G. Robins, and M. Sarrafzadeh an C. K. Wong. Provably Good Performance-Driven Global Routing. *Computer Aided Design*, 11(6):739–752, 1992.

[65] J. Cong, K. S. Leung, and D. Zhou. Performance-Driven Interconnect Design Based on Distributed RC Delay Model. In *ACM Design Automation Conference (DAC'93)*, pages 606–611. ACM, June 1993.

[66] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, 2001.

[67] C. S. Coulston. Constructing Exact Octagonal Steiner Minimal Trees. In *Proceedings of the 13th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 1–6, 2003.

[68] C. S. Coulston. Steiner Minimal Trees in a Hexagonally Partitioned Space. *Internationl Journal of Smart Engineering System Design*, pages 341–346, 2003.

[69] R. Courant and H. Robbins. *What is Mathematics?* Oxford University Press, London, 1941.

[70] S. E. Dreyfus and R. A. Wagner. The Steiner Problem in Graphs. *Networks*, 1:195–207, 1971.

[71] D.-Z. Du, B. Gao, R. L. Graham, Z.-C. Liu, and P.-J. Wan. Minimum Steiner Trees in Normed Planes. *Discrete and Computational Geometry*, 9:351–370, 1993.

[72] D. Z. Du and F. K. Hwang. A Proof of Gilbert and Pollak's Conjecture on the Steiner Ratio. *Algorithmica*, 7:121–135, 1992.

[73] D. Z. Du and F. K. Hwang. Reducing the Steiner Problem in a Normed Space. *SIAM Journal on Computing*, 21:1001–1007, 1992.

[74] W. C. Elmore. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics*, 19:55–63, 1948.

[75] U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of the ACM*, 45(4):634–652, 1998.

[76] Z. Feng, Y. Hu, T. T. Jing, X. L. Hong, X. D. Hu, and G. Y. Yan. An $O(n\ logn)$ Algorithm for Obstacle-avoiding Routing Tree Construction in the λ-geometry Plane. In *Proceedings of the 2006 International Symposium on Physical Design)*, pages 48–55, 2006.

[77] I. Frommer, B. Golden, and G. Pundoor. Heuristic Methods for Solving Euclidean Non-Uniform Steiner Tree Problems. In *The Next Wave in Computing, Optimization, and Decision Technologies*, volume 29 of *Operations Research/Computer Science Interfaces*, pages 133–148. Springer, New York, 2005.

[78] J. L. Ganley and J. P. Cohoon. Routing a Multi-Terminal Critical Net: Steiner Tree Construction in the Presence of Obstacles. In *Proceedings of the International Symposium on Circuits and Systems*, pages 113–116, 1994.

[79] B. Gao, D.-Z. Du, and R. L. Graham. The Tight Lower Bound for the Steiner Ratio in Minkowski Planes. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, pages 183–191, 1994.

[80] M. R. Garey, R. L. Graham, and D. S. Johnson. The Complexity of Computing Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 32(4):835–859, 1977.

[81] M. R. Garey and D. S. Johnson. The Rectilinear Steiner Tree Problem is NP-Complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[82] G. Georgakopoulus and C. H. Christofides. The 1-Steiner Tree Problem. *Journal of Algorithms*, 8:122–130, 1987.

[83] S. H. Gerez. *Algorithms for VLSI Design Automation*. John Wiley & Sons, Inc., New York, USA, 1999.

[84] E. N. Gilbert. Random Minimal Trees. *J. SIAM*, 13:376–387, 1965.

[85] E. N. Gilbert and H. O. Pollak. Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

[86] J. Hu H. Hou and S. S. Sapatnekar. Non-Hanan Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(4):436–444, April 1999.

[87] S. L. Hakimi. Steiner's Problem in Graphs and Its Implications. *Networks*, 1:113–133, 1971.

[88] M. Hanan. On Steiner's Problem with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 14(2):255–265, 1966.

[89] F. C. Harris. Steiner Minimal Trees: An Introduction, Parallel Computation, and Future Work. In D. Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization (Volume 2)*, pages 105–157. Kluwer Academic Publishers, New York, 1998.

[90] M. Hayase. Exact Location of the Steiner Point in the Three-Point Steiner Minimum Tree for λ-geometry. *Electronics and Communications in Japan*, 84:84–94, 2001.

[91] M. Hayase and S. Meki. An Algorithm for Steiner Trees in λ-Geometry. *Transactions of Information Processing Society of Japan (IPSJ Journal)*, 38:677–686, 1997.

[92] S. Heiss. A Path Connection Algorithm for Multi-layer Boards. In *Proceedings of the 5th annual Design Automation Workshop (DAC'68)*, pages 6.1–6.14, New York, NY, USA, 1968. ACM.

[93] C. S. Helvig, G. Robins, and A. Zelikovsky. Improved Approximation Bounds for the Group Steiner Problem. In *Proc. Conference on Design Automation and Test in Europe*, pages 406–413, 1998.

[94] D. Hightower. The Interconnection Problem: A Tutorial. *Computer*, 7:18–32, 1974.

[95] J.-M. Ho, G. Vijayan, and C. K. Wong. New Algorithms for the Rectilinear Steiner Tree Problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(2):185–193, 1990.

[96] T.-Y. Ho, C.-F. Chang, Y.-W. Chang, and S.-J. Chen. Multilevel Full-Chip Routing for the X-Based Architecture. In *Design Automation Conference (DAC)*, pages 597–602, 2005.

[97] F. K. Hwang. On Steiner Minimal Trees with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 30:104–114, 1976.

[98] F. K. Hwang. An $O(n \log n)$ Algorithm for Rectilinear Minimal Spanning Trees. *J. ACM*, 26:177–182, 1979.

[99] F. K. Hwang. A Linear Time Algorithm for Full Steiner Trees. *Operations Research Letters*, 4(5):235–237, 1986.

[100] F. K. Hwang and D. S. Richards. Steiner tree problems. *Networks*, 22:55–89, 1992.

[101] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. Annals of Discrete Mathematics 53. Elsevier Science Publishers, Netherlands, 1992.

[102] M. Igarashi, T. Mitsuhashi, A. Le, S. Kazi, Y.-T. Lin, A. Fujimura, and S. Teig. A Diagonal Interconnect Architecture and its Application to RISC Core Design. In *Proceedings of the IEEE International Solid-State Conference*, pages 460–461, 2002.

[103] E. Ihler. The Rectilinear Class Steiner Tree Problem for Intervals on Two Parallel Lines. *Mathematical Programming*, 63(3):281–296, 1994.

[104] E. Ihler, G. Reich, and P. Widmayer. Class Steiner Trees and VLSI-Design. *Discrete Applied Mathematics*, 90:173–194, 1999.

[105] D. P. Il'yutko. Locally Minimal Trees in n-Normed Spaces. *Mathematical Notes*, 74:619–629, 2003.

[106] A. O. Ivanov and A. A. Tuzhilin. *Minimal Networks: The Steiner Tree Problem and Its Generalizations*. CRC Press, 1994.

[107] V. Jarník and O. Kössler. O minimálních grafeth obeahujících n daných bodú. *Cas. Pest. Mat. a Fys.*, 63:223–235, 1934.

[108] T. Jiang and L. Wang. Computing Shortest Networks with Fixed Topologies. In D.-Z. Du, J. M. Smith, and J. H. Rubinstein, editors, *Advances in Steiner Trees*, pages 39–62. Kluwer Academic Publishers, Boston, 2000.

[109] T. T. Jing, Z. Feng, Y. Hu, X. L. Hong, X. D. Hu, and G. Y. Yan. λ-OAT: λ-geometry Obstacle-avoiding Tree Construction with $O(n\ logn)$ Complexity. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26:2073–2079, 2007.

[110] A. B. Kahng, I. I. Mandoiu, and A. Z. Zelikovsky. Higly Scalable Algorithms for Rectilinear and Octilinear Steiner Trees. In *Proceedings of the Asia-Pacific Design Automation Conferrence*, pages 827–833, 2003.

[111] A. B. Kahng, I. I. Mandoiu, and A. Z. Zelikovsky. Practical Approximations of Steiner trees in Uniform Orientation Metrics. In T. E. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*, chapter 43. Chapman & Hall/CRC, London, 2007.

[112] A. B. Kahng and G. Robins. A New Class of Iterative Steiner Tree Heuristics with Good Performance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(7):893–902, 1992.

[113] A. B. Kahng and G. Robins. *On Optimal Interconnections for VLSI*. Kluwer Academic Publishers, Boston, 1995.

[114] S. Khuller, B. Raghavachari, and N. Young. Balancing Minimum Spanning and Shortest-Path Trees. *Algorithmica*, 14(4):305–321, 1995.

[115] C.-K. Koh. Steiner Problem in Octilinear Routing Model. Master's thesis, National University of Singapore, 1995.

[116] C.-K. Koh and P. H. Madden. Manhattan or non-Manhattan? A Study of Alternative VLSI Routing Architectures. In *Proceedings of the 10th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 47–52, 2000.

[117] B. Korte and J. Vygen. Combinatorial Problems in Chip Design. In M. Grötschel and G. O. H. Katona, editors, *Building Bridges Between Mathematics and Computer Science*, pages 333–368. Springer, Berlin, 2008.

[118] H. W. Kuhn. "Steiner's" Problem Revisited. In G. B. Dantzig and B. C. Eaves, editors, *Studies in Optimization*, volume 10 of *MAA Studies in Mathematics*, pages 52–70. Mathematical Association of America, 1974.

[119] G. Lawlor and F. Morgan. Paired Calibrations Applied to Soap Films, Immiscible Fluids, and Surfaces or Networks Minimizing other Norms. *Pacific Journal of Mathematics*, 166:55–83, 1994.

[120] C. Y. Lee. An Algorithm for Path Connections and its Applications. *IRE Transactions on Electronic Computers*, EC-10:346–365, 1961.

[121] D. T. Lee, T. H. Chen, and C. D. Yang. Shortest rectilinear paths among weighted obstacles. In *SCG '90: Proceedings of the Sixth Annual Symposium on Computational Geometry*, pages 301–310, New York, NY, USA, 1990. ACM.

[122] D. T. Lee and C. F. Shen. The Steiner Minimal Tree Problem in the λ-Geometry Plane. In *ISAAC'96, Lecture Notes in Computer Science 1178*, pages 247–255, 1996.

[123] D. T. Lee, C. F. Shen, and C. L. Ding. On Steiner Tree Problem with 45 Degree Routing. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1680–1683, 1995.

[124] D. T. Lee, C. D. Yang, and T. H. Chen. Shortest Rectilinear Paths among Weighted Obstacles. *International Journal of Computational Geometry and Applications*, 1:109–124, 1991.

[125] K. K. Lee and H. W. Leong. SOAR: A Channel Router for Octilinear Routing Model. In *Proc. IEEE Asia-Pacific Conf. on Circuits and Systems*, pages 346–351, 1992.

[126] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons, England, 1990.

[127] Y. Y. Li, S. K. Cheung, K. S. Leung, and C. K. Wong. On the Steiner Tree Problem in λ_3-Metric. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 1564–1567, 1997.

[128] Y. Y. Li, S. K. Cheung, K. S. Leung, and C. K. Wong. Steiner Tree Constructions in λ_3-Metric. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(5):563–574, 1998.

[129] Y. Y. Li, K. S. Leung, and C. K. Wong. Efficient Heuristics for Orientation Metric and Euclidean Steiner Tree Problems. *Journal of Combinatorial Optimization*, 4:79–98, 2000.

[130] Y. Y. Li, K. S. Leung, and C. K. Wong. Steiner Trees in General Nonuniform Orientations. *Computing*, 66:41–78, 2001.

[131] G.-H. Lin and G. Xue. Reducing the Steiner Problem in an A_3-Geometry Plane. Manuscript, 1998.

[132] G.-H. Lin and G. Xue. The Steiner Tree Problem in λ_4-Geometry Plane. In *ISAAC'98, Lecture Notes in Computer Science 1533*, pages 327–337, 1998.

[133] G.-H. Lin and G. Xue. A Linear Time Algorithm for Computing Hexagonal Steiner Minimum Trees for Terminals on the Boundary of a Regular Hexagon. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 196–199, 2000.

[134] G.-H. Lin and G. Xue. Optimal Layout of Hexagonal Minimum Spanning Trees in Linear Time. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 633–636, 2000.

[135] G.-H. Lin and G. Xue. Reducing the Steiner Problem in Four Uniform Orientations. *Networks*, 35:287–301, 2000.

[136] G.-H. Lin, G. Xue, and D. Zhou. Approximating Hexagonal Steiner Minimal Trees by Fast Optimal Layout of Minimum Spanning Trees. In *International Conference on Computer Design (ICCD)*, pages 392–398, 1999.

[137] G.H. Lin, A. P. Thurber, and G. Xue. The 1-Steiner Tree Problem in λ_3 Geometry Plane. In *Proc. 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 125–128, 1999.

[138] W. Lipski and F. P. Preparata. A Uniform Approach to Layout Wirability. *Math. System Theory*, 19:189–203, 1987.

[139] Z.-C. Liu and D.-Z. Du. On Steiner Minimal Trees with L_p Distance. *Algorithmica*, 7(2/3):179–191, 1992.

[140] E. Lodi. Routing Multiterminal Nets in a Diagonal Model. In *Proceedings of the 1988 Conference on Information Sciences and Systems*, pages 899–902. Department of EE, Princeton University, 1988.

[141] E. Lodi, F. Luccio, and L. Pagli. A Preliminary Study of a diagonal Channel-routing Model. *Algorithmica*, 4:585–597, 1989.

[142] E. Lodi, F. Luccio, and L. Pagli. Routing in Times Square Mode. *Information Processing Letters*, 35:41–48, 1990.

[143] E. Lodi, F. Luccio, and X. Song. A 2D Channel Router for the Diagonal Model. *Integration, the VLSI Journal*, 11:111–125, 1991.

[144] H. Martini, K. J. Swanepoel, and G. Weiss. The Fermat-Torricelli Problem in Normed Planes and Spaces. *Journal of Optimization Theory and Applications*, 115:283–314, 2002.

[145] Z. A. Melzak. On the Problem of Steiner. *Canad. Math. Bull.*, 4(2):143–148, 1961.

[146] Z. A. Melzak. *Companion to Concrete Mathematics*, volume II. John Wiley & Sons, New York, 1976.

[147] R. Möhring, D. Wagner, and F. Wagner. VLSI Network Design. In *Handbooks in Operations Research and Management Science, Vol. 7*, Amsterdam, 1995. Elsevier Science.

[148] M. Müller-Hannemann and S. Peyer. Approximation of Rectilinear Steiner Trees with Length Restrictions on Obstacles. In *WADS 2003: Lecture Notes in Computer Science 2748*, pages 207–218, 2003.

[149] M. Müller-Hannemann, A. Schulze, and L. Zhang. Hardness and Approximation of Octilinear Steiner Trees. *International Journal of Computational Geometry and Applications*, 17:231 – 260, 2007.

[150] M. Müller-Hannemann and S. Tazari. A Near Linear Time Approximation Scheme for Steiner Tree Among Obstacles in the Plane. In *WADS 2007: Lecture Notes in Computer Science 4619*, pages 151–162, 2007.

[151] J. Naor and B. Schieber. Improved Approximations for Shallow-Light Spanning Trees. In *Proc. 38th Ann. Symp. on Foundations of Computer Science*, pages 536–541, 1997.

[152] G. Narasimhan and M. Zachariasen. Geometric Minimum Spanning Trees via Well-Separated Pair Decompositions. *ACM Journal of Experimental Algorithmics*, 6, 2001.

[153] S. Natarajan, N. Sherwani, N. D. Holmes, and M. Sarrafzadeh. Over-the-cell Channel Routing for High Performance Circuits. In *Proceedings of the 29th ACM/IEEE Design Automation Conference (DAC'92)*, pages 600–603, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.

[154] B. K. Nielsen, P. Winter, and M. Zachariasen. An Exact Algorithm for the Uniformly-Oriented Steiner Tree Problem. In *Proceedings of the 10th*

European Symposium on Algorithms, Lecture Notes in Computer Science, volume 2461, pages 760–772. Springer, 2002.

[155] B. K. Nielsen, P. Winter, and M. Zachariasen. On the Location of Steiner Points in Uniformly-Oriented Steiner Trees. *Information Processing Letters*, 83:237–241, 2002.

[156] B. K. Nielsen, P. Winter, and M. Zachariasen. Rectilinear Trees under Rotation and Related Problems. In *Proceedings of the 18th European Workshop on Computational Geometry*, pages 18–22, 2002.

[157] M. H. Pagh. Steiner Trees in Weighted Fixed Orientation Metrics. Master's thesis, Department of Computer Science, University of Copenagen, 2005.

[158] M. Paluszewski. VLSI Routing in Uniform Orientation Metrics. Master's thesis, Department of Computer Science, University of Copenhagen, 2004.

[159] M. Paluszewski, P. Winter, and M. Zachariasen. A New Paradigm for General Architecture Routing. In *Proceedings of the 14th ACM Great Lakes symposium on VLSI (GLSVLSI)*, pages 202–207, 2004.

[160] M. Pecht and Y. T. Wong. *Advanced Routing of Electronic Modules*. CRC Press, New York, 1996.

[161] S. Peyer. Elmore-Delay-optimale Steinerbäume im VLSI-Design. Master's thesis, Research Institute for Discrete Mathematics, University of Bonn, 2000.

[162] S. Peyer. *Shortest Paths and Steiner Trees in VLSI Routing*. PhD thesis, Research Institute for Discrete Mathematics, University of Bonn, 2007.

[163] S. Peyer, M. Zachariasen, and D. G. Jørgensen. Delay-related Secondary Objectives for Rectilinear Steiner Minimum Trees. *Discrete Applied Mathematics*, 136:271–298, 2004.

[164] T. Polzin and S. Vahdati Daneshmand. Improved Algorithms for the Steiner Problem in Networks. *Discrete Applied Mathematics*, 112:263–300, 2001.

[165] T. Polzin and S. Vahdati Daneshmand. Approaches to the Steiner Problem in Networks. *Lecture Notes in Computer Science*, 5515:81–103, 2009.

[166] K. Powers, D. Brown, and M. L. Brady. The 60° Grid: Routing Channals in Width $d/\sqrt{3}$. In *Proceedings of the 1st Great Lakes Symp. on VLSI*, pages 214–219, 1991.

[167] F.P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, second edition, 1988.

[168] H. J. Prömel and A. Steger. *The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity*. Advanced Lectures in Mathematics. Friedrick Vieweg & Son, 2002.

[169] J. S. Provan. An Approximation Scheme for Finding Steiner Trees with Obstacles. *SIAM Journal on Computing*, 17(5):920–934, 1988.

[170] S. B. Rao and W. D. Smith. Approximating Geometrical Graphs via "Spanners" and "Banyans". In *Proc. Thirtieth Annual ACM Symposium on Theory of Computing*, pages 540–550, 1998.

[171] D. Rautenbach. Rectilinear Spanning Trees versus Bounding Boxes. *The Electronic Journal of Combinatorics*, 11:N12, 2004.

[172] G. Reich and P. Widmayer. Beyond Steiner's Problem: A VLSI Oriented Generalization. In *Proc. 15th Internat. Workshop Graph-Theoret. Concepts Comput. Sci., Lecture Notes in Computer Science*, volume 411, pages 196–210. Springer, 1990.

[173] G. Robins and A. Zelikovsky. Improved Steiner Tree Approximation in Graphs. In *SODA '00: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779. Society for Industrial and Applied Mathematics, 2000.

[174] J. H. Rubinstein and D. A. Thomas. A Variational Approach to the Steiner Network Problem. *Annals of Operations Research*, 33:481–499, 1991.

[175] S. M. Sait and H. Youssef. *VLSI Physical Design Automation: Theory and Practice*. World Scientific Publishing, Singapore, 1999.

[176] D. Sankoff and P. Rousseau. Locating the Vertices of a Steiner Tree in an Arbitrary Metric Space. *Mathematical programming*, 9:240–246, 1975.

[177] M. Sarrafzadeh. *Hierarchical Approaches to VLSI Circuit Layout*. PhD thesis, University of Illinois at Urbana-Champaign, 1986.

[178] M. Sarrafzadeh, W.-L. Lin, and C. K. Wong. Floating Steiner Trees. *IEEE Transactions on Computers*, 47(2):197–211, 1998.

[179] M. Sarrafzadeh and C. K. Wong. Bottleneck Steiner Trees in the Plane. *IEEE Transactions on Computers*, 41:370–374, 1992.

[180] M. Sarrafzadeh and C. K. Wong. Hierarchical Steiner Tree Construction in Uniform Orientations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11:1095–1103, 1992.

[181] M. Sarrafzadeh and C.K. Wong. *An Introduction to VLSI Physical Design*. McGraw-Hill, New York, 1996.

[182] P. Saxena, R. S. Shelar, and S. Sapatnekar. *Routing Congestion in VLSI Circuits: Estimation and Optimization*. Springer, New York, 2007.

[183] A. Schulze. *Approximation Algorithms for Network Design Problems*. PhD thesis, Universität zu Köln, 2008.

[184] C. F. Shen. *The λ-geometry Steiner Minimal Tree Problem and Visualization*. PhD thesis, Northwestern University, Evanston, IL, USA, 1997.

[185] N. A. Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, Boston, third edition, 1999.

[186] A. Shulze and M. Müller-Hannemann. Hardness and Approximation of Octilinear Steiner Trees. In *Proceedings of The 16th Annual International Symposium on Algorithms and Computation (ISAAC)*, pages 256–265, 2005.

[187] J. M. Smith, D. T. Lee, and J. S. Liebman. An $O(n \log n)$ Heuristic for Steiner Minimal Tree Problems on the Euclidean Metric. *Networks*, 11:23–29, 1981.

[188] W. D. Smith. How to Find Steiner Minimal Trees in Euclidean d-Space. *Algorithmica*, 7(2/3):137–177, 1992.

[189] T. L. Snyder. On the Exact Location of Steiner Points in General Dimension. *SIAM Journal on Computing*, 21(1):163–180, 1992.

[190] X. Song and X. Tan. An Optimal Channel-Routing Algorithm in the Times Square Model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13:891–998, 1994.

[191] M. Stan, F. Hamzaoglu, and D. Garrett. Non-manhattan maze routing. In *Proceedings of the 17th Symposium on Integrated Circuits and System Design (SBCCI'04)*, pages 260–265, New York, NY, USA, 2004. ACM.

[192] K. J. Swanepoel. Vertex Degrees of Steiner Minimal Trees in l_p^d and Other Smooth Minkowski Spaces. *Discrete Comput. Geom.*, 21:437–447, 1999.

[193] K. J. Swanepoel. The Local Steiner Problem in Normed Planes. *Networks*, 36:104–113, 2000.

[194] K. J. Swanepoel. The Local Steiner Problem in Finite-Dimensional Normed Spaces. *Discrete and Computational Geometry*, 37:419–442, 2007.

[195] X. Tan and X. Song. Hexagonal 3-Layer Channel Routing. *Information Processing Letters*, 55:223–228, 1995.

[196] X. Tan and X. Song. Routing Multiterminal Nets on an Hexagonal Grid. *Discrete Applied Mathematics*, 90:245–255, 1999.

[197] S. Teig. The X Architecture. In *Proceedings ACM/IEEE Workshop on System Level Interconnect Prediction*, pages 33–37, 2002.

[198] S. Teig. The X Architecture: Not Your Father's Diagonal Wiring. *International Workshop on System-Level Interconnect Prediction (SLIP)*, pages 33–37, 2002.

[199] A. P. Thurber and G. Xue. Computing Hexagonal Steiner Trees using PCx. In *Proc. 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 381–384, 1999.

[200] I. G. Tollis. Techniques for Wiring in Non-Square Grids. In *IEEE International Symposium on Circuits and Systems '89*, pages 66–69, 1989.

[201] I. G. Tollis. Wiring in Uniform Grids and Two-Colorable Maps. *Integration, the VLSI Journal*, 12:189–210, 1991.

[202] E. Uchoa, M. Poggi de Aragão, and C. Ribeiro. Preprocessing Steiner Problems from VLSI Layout. *Networks*, 40:38–50, 2002.

[203] A. Underwood. A Modified Melzak Procedure for Computing Node-Weighted Steiner Trees. *Networks*, 27:73–79, 1996.

[204] J. Vygen. *Theory of VLSI Layout*. Habilitation, University of Bonn, 2001.

[205] D. C. Wang. Novel Schemes for IC Layout, Part I: Two-Layer Channel Routing. In *Proc. 28th ACM/IEEE Design Automation Conf.*, pages 49–53, 1991.

[206] D. M. Warme. *Spanning Trees in Hypergraphs with Applications to Steiner Trees*. PhD thesis, Computer Science Dept., The University of Virginia, 1998.

[207] D. M. Warme, P. Winter, and M. Zachariasen. Exact Solutions to Large-Scale Plane Steiner Tree Problems. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 979–980, 1999.

[208] D. M. Warme, P. Winter, and M. Zachariasen. Exact Algorithms for Plane Steiner Tree Problems: A Computational Study. In D.-Z. Du, J. M. Smith, and J. H. Rubinstein, editors, *Advances in Steiner Trees*, pages 81–116. Kluwer Academic Publishers, Boston, 2000.

[209] D. M. Warme, P. Winter, and M. Zachariasen. GeoSteiner 3.1. Department of Computer Science, University of Copenhagen (DIKU), http://www.diku.dk/geosteiner/, 2001.

[210] Y. Wei, S. Dong, and X. Hong. APWL-Y: An Accurate and Efficient Wire-length Estimation Technique for Hexagon/Triangle Placement. *Integration, the VLSI Journal*, 40:406–419, 2007.

[211] E. Welzl. Constructing the Visibility Graph for n-line Segments in $O(n^2)$ Time. *Information Processing Letters*, 20:167–171, 1985.

[212] T. Whitney and C. Mead. An Integer Based Hierarchical Representation for VLSI. In *Advanced Research in VLSI (Proc. 4th MIT Conf.)*, pages 241–257, 1986.

[213] P. Widmayer. On Graphs Preserving Rectilinear Shortest Paths in the Presence of Obstacles. *Annals of Operations Research*, 33:557–575, 1991.

[214] P. Widmayer, Y. F. Wu, and C. K. Wong. Distance Problems in Computational Geometry with Fixed Orientations. In *Proceedings of the Symposium on Computational Geometry, Baltimore, MD*, pages 186–195, 1985.

[215] P. Widmayer, Y. F. Wu, and C. K. Wong. On Some Distance Problems in Fixed Orientations. *SIAM Journal on Computing*, 16(4):728–746, 1987.

[216] P. Winter. An Algorithm for the Steiner Problem in the Euclidean Plane. *Networks*, 15:323–345, 1985.

[217] P. Winter. Euclidean Steiner Minimal Trees with Obstacles and Steiner Visibility Graphs. *Discrete Applied Mathematics*, 47:187–206, 1993.

[218] P. Winter. Reductions for the Rectilinear Steiner Tree Problem. *Networks*, 26:187–198, 1995.

[219] P. Winter and J. M. Smith. Steiner Minimal Trees for Three Points with One Convex Polygonal Obstacle. *Annals of Operations Research*, 33:577–599, 1991.

[220] P. Winter and M. Zachariasen. Euclidean Steiner Minimum Trees: An Improved Exact Algorithm. *Networks*, 30:149–166, 1997.

[221] Y. F. Wu, P. Widmayer, M. D. F. Schlag, and C. K. Wong. Rectilinear Shortest Paths and Minimum Spanning Trees in the Presence of Rectilinear Obstacles. *IEEE Transactions on Computers*, 36:321–331, 1987.

[222] X Initiative Home Page. `http://www.xinitiative.com`, 2001.

[223] G. Xue and D.-Z. Du. An $O(n \log n)$ Average Time Algorithm for Computing the Shortest Network under a Given Topology. *Algorithmica*, 23:354–362, 1999.

[224] G. Xue and K. Thulasiraman. Computing the Shortest Network under a Fixed Topology. *IEEE Transactions on Computers*, 51:1117–1120, 2002.

[225] G. Y. Yan, A. Albrecht, G. H. Yound, and C. K. Wong. The Steiner Tree Problem in Orientation Metrics. *Journal of Computer and System Sciences*, 55:529–546, 1997.

[226] J.-T. Yan. An Improved Optimal Algorithm for Bubble-Sorting-Based Non-Manhattan Channel Routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18:163–171, 1999.

[227] M. C. Yildiz and P. H. Madden. Preferred Direction Steiner Trees. In *Proceedings of the 11th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 56–61, 2001.

[228] M. C. Yildiz and P. H. Madden. Preferred Direction Steiner Trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1368–1372, 2002.

[229] M. Zacharias. *Encyklopädie der Mathematischen Wissenschaften, Article III AB 9*. Teubner, Leipzig, 1914.

[230] M. Zachariasen. Local Search for the Steiner Tree Problem in the Euclidean Plane. *European Journal of Operational Research*, 119:282–300, 1999.

[231] M. Zachariasen. Rectilinear Full Steiner Tree Generation. *Networks*, 33:125–143, 1999.

[232] M. Zachariasen. A Catalog of Hanan Grid Problems. *Networks*, 38:76–83, 2001.

[233] M. Zachariasen. The Rectilinear Steiner Tree Problem: A Tutorial. In D.-Z. Du and X. Cheng, editors, *Steiner Trees in Industries*, pages 467–507. Kluwer Academic Publishers, Boston, 2001.

[234] M. Zachariasen. Comment on "Computing the Shortest Network under a Fixed Topology". *IEEE Transactions on Computers*, 55:783–784, 2006.

[235] M. Zachariasen and A. Rohe. Rectilinear Group Steiner Trees and Applications in VLSI Design. *Mathematical Programming*, 94:407–433, 2003.

[236] M. Zachariasen and P. Winter. Concatenation-Based Greedy Heuristics for the Euclidean Steiner Tree Problem. *Algorithmica*, 25:418–437, 1999.

[237] M. Zachariasen and P. Winter. Obstacle-Avoiding Euclidean Steiner Trees in the Plane: An Exact Algorithm. In *Workshop on Algorithm Engineering and Experimentation (ALENEX), Lecture Notes in Computer Science 1619*, pages 282–295. Springer, 1999.

[238] A. Z. Zelikovsky. An 11/6-Approximation for the Network Steiner Tree Problem. *Algorithmica*, 9:463–470, 1993.

[239] Q. Zhu, H. Zhou, T. Jing, X. Hong, and Y. Yang. Efficient Octilinear Steiner tree Construction based on Spanning Graphs. In *Proceedings of the 2004 Conference on Asia South Pacific Design Automation (ASP-DAC)*, pages 687–690, 2004.

Summary (in Danish)

I forbindelse med design af integrerede kredsløb (chips) indgår der en række såkaldte forbindelsesproblemer. En moderne chip består af flere milliarder transistorer, som skal forbindes med metalledninger på chippens overflade. Disse metalledninger lægges i et (lille) antal lag, således at uafhængige elektriske net ikke overlapper med hinanden. Den traditionelle fremstillingsteknologi kan kun håndtere horisontale og vertikale forbindelser på chippens overflade — og bliver betegnet Manhattan-arkitektur.

De seneste 10 år har interessen for generelle arkitekturer, hvor mere end to orienteringer kan benyttes til at forbinde transistorerne, været stigende. Denne udvikling har resulteret i en betydelig forskning i forbindelsesproblemer med faste (men ellers vilkårlige) orienteringer. Minimering af forbindelsernes længde — det såkaldte Steiner problem med faste orienteringer — har været genstand for særlig opmærksomhed.

Denne doktorafhandling består af 12 forskningsartikler samt en oversigtsartikel om Steiner problemet med faste orienteringer — med nogle af dets generaliseringer. Et af hovedbidragene er en lineær-tids algoritme, der kan konstruere et minimalt Steiner træ for en given topologi. Desuden vises, at samme problem kan løses ved hjælp af lineær programmering. For det generelle problem, hvor topologien er ukendt, præsenteres en eksakt algoritme, der kan løse problemet med flere tusinde punkter til optimalitet. Der præsenteres et nyt paradigma for konstruktion af netforbindelser på en chip under en generel arkitektur med faste orienteringer. Resultaterne dokumenterer, at der er klare fordele ved at benytte mere end to orienteringer i chip design.

Afhandlingen afsluttes med en beskrivelse af en række generaliseringer af Steiner problemet, der udspringer fra chip design. Der præsenteres et katalog af problemer, som kan løses på det såkaldte Hanan gitter. Desuden behandles generaliseringer, som kan håndtere signalforsinkelser og gruppeforbindelser. Til sidst gives en række egenskaber for Steiner problemet med tilladt rotation af de faste orienteringer.

Resultaterne udgør et væsentligt teoretisk og algoritmisk bidrag til forståelsen af Steiner problemet med faste orienteringer. Desuden fokuserer afhandlingen på anvendelser i chip design.

Research papers

[A] M. Brazil, B. K. Nielsen, P. Winter, and M. Zachariasen. Rotationally Optimal Spanning and Steiner Trees in Uniform Orientation Metrics. *Computational Geometry: Theory and Applications*, 29:251–263, 2004.

[B] M. Brazil, D. A. Thomas, J. F. Weng, and M. Zachariasen. Canonical Forms and Algorithms for Steiner Trees in Uniform Orientation Metrics. *Algorithmica*, 44:281–300, 2006.

[C] M. Brazil, P. Winter, and M. Zachariasen. Flexibility of Steiner Trees in Uniform Orientation Metrics. *Networks*, 46:142–153, 2005.

[D] M. Brazil and M. Zachariasen. Steiner Trees for Fixed Orientation Metrics. *Journal of Global Optimization*, 43:141–169, 2009.

[E] G. Narasimhan and M. Zachariasen. Geometric Minimum Spanning Trees via Well-Separated Pair Decompositions. *ACM Journal of Experimental Algorithmics*, 6, 2001.

[F] B. K. Nielsen, P. Winter, and M. Zachariasen. An Exact Algorithm for the Uniformly-Oriented Steiner Tree Problem. In *Proceedings of the 10th European Symposium on Algorithms, Lecture Notes in Computer Science*, volume 2461, pages 760–772. Springer, 2002.

[G] B. K. Nielsen, P. Winter, and M. Zachariasen. On the Location of Steiner Points in Uniformly-Oriented Steiner Trees. *Information Processing Letters*, 83:237–241, 2002.

[H] M. Paluszewski, P. Winter, and M. Zachariasen. A New Paradigm for General Architecture Routing. In *Proceedings of the 14th ACM Great Lakes symposium on VLSI (GLSVLSI)*, pages 202–207, 2004.

[I] S. Peyer, M. Zachariasen, and D. G. Jørgensen. Delay-related Secondary Objectives for Rectilinear Steiner Minimum Trees. *Discrete Applied Mathematics*, 136:271–298, 2004.

[J] M. Zachariasen. A Catalog of Hanan Grid Problems. *Networks*, 38:76–83, 2001.

[K] M. Zachariasen. Comment on “Computing the Shortest Network under a Fixed Topology”. *IEEE Transactions on Computers*, 55:783–784, 2006.

[L] M. Zachariasen and A. Rohe. Rectilinear Group Steiner Trees and Applications in VLSI Design. *Mathematical Programming*, 94:407–433, 2003.

ISBN-nr: 978-87-981270-4-8